

Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click

Robert Xiao Gierad Laput Chris Harrison

Human-Computer Interaction Institute

Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213

{brx, gierad.laput, chris.harrison}@cs.cmu.edu

ABSTRACT

Smartwatches promise to bring enhanced convenience to common communication, creation and information retrieval tasks. Due to their prominent placement on the wrist, they must be small and otherwise unobtrusive, which limits the sophistication of interactions we can perform. This problem is particularly acute if the smartwatch relies on a touchscreen for input, as the display is small and our fingers are relatively large. In this work, we propose a complementary input approach: using the watch face as a multi-degree-of-freedom, mechanical interface. We developed a proof of concept smartwatch that supports continuous 2D panning and twist, as well as binary tilt and click. To illustrate the potential of our approach, we developed a series of example applications, many of which are cumbersome – or even impossible – on today’s smartwatch devices.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces - Input devices and strategies.

Keywords: Wearable computing; buttons; touchscreens; watch; smart clothing; on-body interfaces.

INTRODUCTION

The wristwatch has been an ever-present wearable technology since its inception in the early 1900s, and has undergone continuous technical refinement since that time. Researchers have long viewed the immediacy and ubiquity of the wristwatch as a vehicle for computation, pushing its capabilities to ever-greater heights. In 2000, IBM demonstrated the first watch running a full operating system [20]. However, unlike smartphones, which can be scaled to a variety of sizes, smartwatches must be small and unobtrusive in order to remain socially acceptable, which has long limited their practicality.

Recently, smartwatches have experienced a resurgence of interest as electronics have become more powerful and power efficient, making them more practical and capable than ever before. Products like the Pebble smartwatch have been popularized in the public press, and prominent manufacturers such as Qualcomm, Sony, Motorola and Samsung

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2014, April 26 - May 01 2014, Toronto, ON, Canada

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2473-1/14/04 \$15.00. <http://dx.doi.org/10.1145/2556288.2557017>

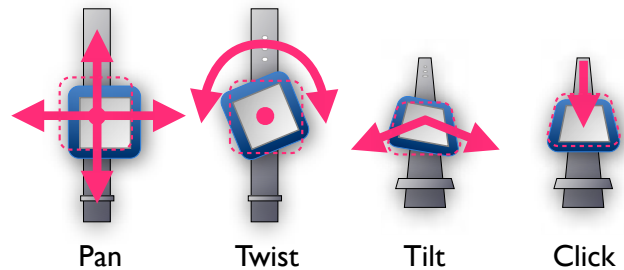


Figure 1. Our smartwatch prototype supports continuous 2D panning and twist, along with binary tilt and click.

have each recently released their own smartwatches. These devices rely on small buttons and/or touchscreens for input.

We propose using the watch face as a multi-degree-of-freedom mechanical interface. Our proof-of-concept smartwatch supports continuous 2D panning and twist, as well as binary tilt and click (Figures 1 and 2). This input approach can operate in parallel with buttons, touchscreens and voice, offering greater expressivity in the same compact form.

PROOF OF CONCEPT IMPLEMENTATION

Our prototype, seen in Figure 2, has four primary components. The topmost element is a 1.5-inch TFT LCD color display with a resolution of 280 x 220 pixels. This display is mounted on top of a pair of N35P112 Hall-effect joystick sensors, set apart by 35mm (attached to the left and right bezel of the display). These sensors capture 2D movements [23] with a maximum displacement of 0.5mm. Additionally, each sensor has a binary click action perpendicular to the movement plane.

This configuration allows several different manipulations of the display to be digitized (Figure 1). Specifically, panning the display (i.e., shear or tangential motion [5,6]) results in identical X/Y translations on the joysticks underneath. Analog twist (i.e., clockwise/counterclockwise motions) can be calculated by looking for opposing translations on the joysticks [11]. When a single sensor is depressed, we interpret



Figure 2. Our prototype uses a pair of Hall-effect joystick sensors, providing a multi-degree-of-freedom interface.

this as a binary tilt action (i.e., tilted left or right). When both sensors are depressed, a click event is fired.

The two sensors are connected to an ARM Cortex M3-based Maple Mini over an I²C bus. Each sensor provides 8-bit X and Y position data, as well as click state, 200 times per second. This data is sent via USB to a host computer, where a Java program processes the inputs and implements an interactive application. The graphical output of this application is sent to the smartwatch display.

A more advanced prototype could include additional input dimensions. For example, instead of joystick sensors with binary click, pressure sensors could be used. This would allow for analog tilt sensing. With four such sensors, 2D analog tilt is possible – yielding a device with pitch, yaw and roll, in addition to 2D positional tracking.

RELATED WORK

Small devices have appeared in many forms, ranging from “inch-scale” devices [24], to digital jewelry [12], and now smartwatches. The latter form has inspired a myriad of research efforts, of which we highlight a few (see [7] for a more complete overview).

Touchscreens have been used in various ways to enhance interactions on wearable and diminutive devices. For example, Facet [10] was a prototype smartwatch that used many small touchscreens synergistically to enable more sophisticated operations. Zoomboard [16] used iterative zooming to make text entry on smartwatch-sized displays more accurate. NanoTouch [2] moved the touchpad to the rear of the device so that interactions do not occlude the screen.

In-air gestures can utilize area around the watch for input with minimal screen occlusion. For example, the Gesture Watch [8] and HoverFlow [9] used IR proximity sensors to capture gestures performed above a device. Ni and Baudisch [15] used a tiny, wrist-worn, high-resolution optical sensor to recognize gestures performed on top of it. Abracadabra [4] used free-space magnetic tracking to sense an instrumented finger near a wrist-worn device. However, these gesture-based techniques require specific clutching mechanisms, lack tactile feedback, and are generally indiscreet. Our approach hopes to mitigate these issues by providing users a physical, direct manipulation interface.

Utilizing the watchband or the bezel for input is another popular technique, as it minimizes screen occlusion. WatchIt [18] instrumented the watchband itself to act as a simple slider or button input. Ashbrook et al. [1] proposed using a capacitive bezel around the screen as an additional input channel. Blasko et al. [3] used raised, touch-sensitive segments around the bezel to enable strokes and taps. Pasquero et al. [17] combined a rotatable bezel and touchscreen with haptic feedback to create an eyes-free watch interface. Although these techniques mitigate occlusion, they may require larger surface areas to operate effectively, and generally only offer 1D radial input. In contrast, our approach uses hall-effect sensors operating behind the watchface and provides three or more analog degrees of freedom.



Figure 3. Panning the watch moves the map, twisting zooms in and out, and clicking shows details.

Also related to our technique are systems that sense physical force applied to a screen. Touchscreens with shear capabilities [5,6] can sense the tangential forces of touch inputs, offering an additional input dimension. Tilt has also been explored for mobile input [21]. Lastly, the use of pressure as an input channel has long been studied for applications such as continuous scrolling and zooming [13,19].

INTERACTION TECHNIQUES

To illustrate the potential and immediate feasibility of our approach, we developed seven applications to showcase five interaction techniques we developed on our prototype hardware (Figures 3-9 and Video Figure). Of note, we reserve left and right tilt for system-wide functionality, such as switching between applications. Thus, our applications only utilize pan, twist and click inputs for interactive use.

Pan + Zoom

Our smartwatch offers 2D panning and zooming (via twist), which is ideal for the navigation of planar content [14]. Using this interaction paradigm, we created a map application (Figure 3). Tangentially pushing the watch face pans the map; users can vary the applied force to move at different speeds. A cursor is shown in the center of the screen; clicking the smartwatch displays information on the location beneath the cursor. This modality could also be used for navigation (e.g., webpages, eBooks and photos).

Twist + Click

Many applications require only a single degree of freedom for navigation, coupled with a binary click for action or selection. For example, we created a clock application that displays the current time (Figure 4). When the watch face is depressed (i.e., clicked), the clock switches mode, allowing the user to set an alarm time. Twisting the watch clockwise and counterclockwise adjusts the alarm time. As our prototype provides analog twist, the application offers rate-controlled movement, allowing for rapid and precise time



Figure 4. Clicking and then twisting sets the alarm time.



Figure 5. In our camera application, twisting zooms the camera and clicking snaps a photo.

selection. Another application that can use a twist+click interaction scheme is a camera (Figure 5). The application displays a live preview from a camera mounted on the wrist strap. Twisting the watch face zooms the camera in and out; clicking snaps a picture.

Four-Way Navigation

To illustrate this interaction technique, we created a music player (Figure 6), which naturally lends itself to a hierarchical menu structure. Users pan the watch right to select menu items, and left to return to the previous menu. Within menus, panning up and down scrolls the list at a variable rate. At the topmost level, users select a categorization (albums, artists, genres and playlists), then a specific group of songs within that categorization. They can then select a song to play, and change the currently playing song using up-and-down panning motions. Additionally, twist and click inputs are recognized application-wide: twist adjusts the volume, while clicks pause and play the current song.

Calendars are another popular application for smartwatches. In our calendar application (Figure 7), panning left or right allows the user to move between days, while panning up or down allows the user to view calendar items for a particular day. Clicking on an event brings up additional detail.

Gestures

Another interaction technique is to manipulate the watch face gesturally. For example, the user can grasp the display and move it in a circular fashion. Other shapes are possible, including triangles, and also characters, such as ‘V’, ‘^’ and ‘[’. As a proof of concept, we created a basic application utilizing the \$1 Gesture Recognizer [25] using 2D pan position (Figure 8). The application supported all of the aforementioned gestures, and could be used in the future for quick actions such as turning the device to silent, playing voicemails, taking a photo, and bookmarking locations.

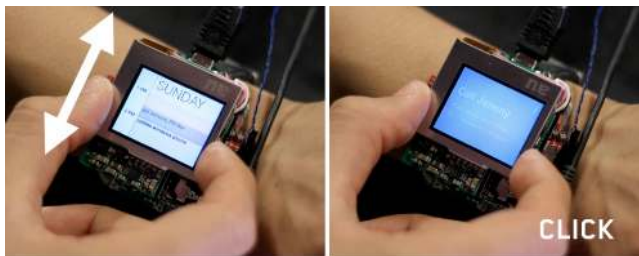


Figure 7. In our calendar app, panning moves between calendar items; clicking displays a detail view.

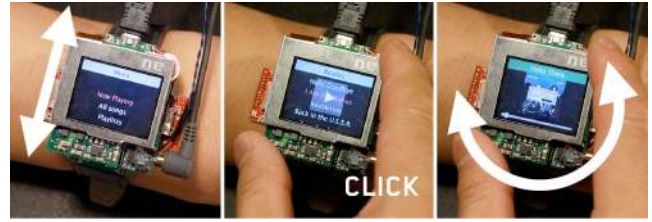


Figure 6. In the music application, panning navigates menus, twisting adjusts the volume, and clicking plays or pauses.

Gamepad Input

Games can often take advantage of expressive input dimension, such as those offered by our prototype. To demonstrate the potential for smartwatch gaming, we ported Doom, a first-person shooter by id Software (Figure 9). 2D pan is bound to forward/backward movement and left/right strafing. Twist is used to turn left and right, and click shoots. Other control schemes are possible. For example, a racing game could utilize twist for steering wheel control, which is ideal, as it offers continuous, analog control. Depressing the watch face could be coupled to the gas pedal. Finally, 2D games such as Tetris, Space Invaders, and more generally, side-scrollers and overheads, could use panning for up/down/left/right player movement, while twist and click could provide different actions, (e.g., changing block orientations, selecting weapons, shooting).

DRAWBACKS AND LIMITATIONS

The approach we present complements contemporary smartwatch input mechanisms, namely touch, physical buttons and voice. Our approach does not hinder any of latter modalities, while simultaneously enabling new dimensions of input, which we view as a significant benefit. That being said, there are several drawbacks and limitations that should also be noted, which we now describe.

Foremost, our approach requires extra components to be placed inside of a smartwatch. This is no small feat given the compactness of these devices. Nonetheless, our prototype was relatively small and used off-the-shelf components). A tightly integrated solution should be readily possible in the near future. In addition to requiring extra space inside of the device, the sensors we require will also increase cost and potentially power consumption.

Secondly, mechanical components inside mobile devices are less than ideal (the industry has almost entirely moved to solid state, save perhaps buttons). This is because particulates and water have the potential to enter the enclosure and wear and even destroy these components. While gaskets can be employed to seal joints with mechanical movement, they



Figure 8. Gestures can be input by clicking and panning.

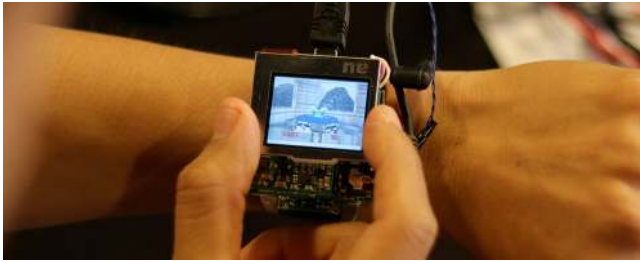


Figure 9. Doom running on our smartwatch prototype.

are not 100% effective, and also increase cost and complexity in manufacturing. One possible alternative is to use strain gauges, force sensitive resistors, and similar sensors, which measure the same forces we employ, but with essentially zero mechanical displacement. This would allow tighter integration at the expense of tactile feedback (e.g., twist would really only be torque).

Another issue is the fact that manipulating the watch face displaces the visual content one is trying to interact with. Initially, this was a chief concern, but in practice, we found the effect to be minimal. Fortunately, when users are manipulating the watch face, their visual and motor systems work in tandem, and the perceived effect is minimal [22]. This is why, for example, it is possible for people to read a smartphone while walking, while those around them cannot.

As with capacitive touch, buttons and voice, there is a danger of accidental input. This might occur by snagging on clothing or being “pushed” when e.g., the hands are resting on a table. Like other modalities, an “unlock” mechanism might have to be employed to reduce false activations. Indeed, a device could have one such mechanism shared by all input modalities.

Finally, we learned that with mechanical input, we lose affordances that a touchscreen button can provide in the form of an icon or text. Specifically, untrained users will initially find it difficult to deduce the mapping between mechanical inputs and actions on their watches. However, we observed that people quickly learned these mappings (primarily through trial and error). Similarly, we found that using gestures was particularly difficult without some sort of immediate feedback to visualize the displacement of the watch face. To address this, we added a small dot and ghosted trail (Figure 8) that shows the displacement state and history.

CONCLUSION

In this work, we designed and implemented a new smartwatch interaction modality that uses the watch face as a multi-degree-of-freedom mechanical interface, enabling users to pan, zoom, navigate, select and perform other actions using simple physical controls. We demonstrated these capabilities with a suite of example applications, built around five interaction techniques. Our approach is inexpensive, potentially compact, and can complement existing inputs, such as physical buttons, touch and voice.

REFERENCES

1. Ashbrook, D., Lyons, K. and Starner, T. An investigation into round touchscreen wristwatch interaction. In *Proc. of MobileHCI '08*. 311-314.
2. Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proc. CHI '09*. 1923-1932.
3. Blasko, G. and Feiner, S. An interaction system for watch computers using tactile guidance and bidirectional segmented strokes. In *Proc. ISWC '04*. 120-123.
4. Harrison, C. and Hudson, S. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proc. UIST '09*. 121-124.
5. Harrison, C. and Hudson, S. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proc. CHI '12*. 3149-3152.
6. Heo, S. and Lee, G. Force gestures: augmenting touch screen gestures with normal and tangential forces. In *Proc. UIST '11*. 621-626.
7. Hinckley K. and Song, H. Sensor synaesthesia: touch in motion, and motion in touch. In *Proc. CHI '11*. 801-810.
8. Kim, J., He, J., Lyons, K. and Starner, T. The Gesture Watch: a wireless contact-free gesture based wrist interface. In *Proc. ISWC '07*. 11-13.
9. Kratz, S. and Rohs, M. Hoverflow: exploring around-device interaction with IR distance sensors. *Proc. MobileHCI '09*. 1-4.
10. Lyons, K., Nguyen, D., Ashbrook, D. and White, S.. Facet: a multi-segment wrist worn system. In *Proc. UIST '12*. 123-130.
11. Masui, T., Tsukada, K. and Siio, I. MouseField: a simple and versatile input device for ubiquitous computing. In *Proc. UbiComp '04*. 319-328.
12. Miner, C. S., Chan, D. M. and Campbell, C. Digital jewelry: wearable technology for everyday life. In *CHI EA '01*. 45-46.
13. Miyaki, T. and Rekimoto, J. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In *Proc. MobileHCI '09*. Article 11.
14. Mooser, J., You, S. and Neumann, U. Large document, small screen: a camera driven scroll and zoom control for mobile devices. In *Proc. 13D '08*. 27-34.
15. Ni, T. and Baudisch, P. Disappearing mobile devices. In *Proc. UIST '09*. 101-110.
16. Oney, S., Harrison, C., Ogan, A. and Wiese, J. ZoomBoard: a diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In *Proc. CHI '13*. 2799-2802.
17. Pasquero, J., Stobbe, S. and Stonehouse, N. A haptic wristwatch for eyes-free interactions. In *Proc. CHI '11*. 3257-3266.
18. Perrault, S., Lecolinet, E., Eagan, J. and Guiard, Y. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proc. CHI '13*. 1451-1460.
19. Quinn, P. and Cockburn, A. Zoofing!: faster list selections with pressure-zoom-flick-scrolling. In *Proc. OZCHI '09*. 185-192.
20. Raghunath, M. T. and Narayanaswami, C. User interfaces for applications on a wrist watch. *Personal Ubiquitous Computing*, 6 (January 2002), 17-30.
21. Rahman, M., Gustafson, S., Irani P. and Subramanian, S. Tilt techniques: investigating the dexterity of wrist-based input. In *Proc. CHI '09*. 1943-1952.
22. Ren, L. and Crawford, J. Coordinate transformations for hand-guided saccades. *Exp. Brain Research*. 195 (3), 2009. 455-465.
23. Selker, T. and Rutledge, J. Finger Force Precision for Computer Pointing. IBM RC 17342, October 1991.
24. Weiser, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (July 1999), 3-11.
25. Wobbrock, J., Wilson, A. and Li, Y. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In *Proc. UIST '07*. 159-168.