TouchBand: A Pressure-Sensitive Wristband as Input for Smartwatch Scrolling and Zooming

| Serena Jeblee, Dina Sabie, Shamama Khattak, Gurleen Kaur  Department of Computer Science, University of Toronto  Toronto, Canada  sjeblee@cs.toronto.edu  {dina.sabie, shamama.khattak, gurleen.kaur}@mail.utoronto.ca |
| --- |

# ABSTRACT

Smartwatches popularity is in rise due to their wearablity aspect and the various functions they offer. However, due to the limited screen size of smartwatch’s screen, there are fewer touchscreen gestures available for smartwatches than for smartphones. For example, pinching and stretching fingers to zoom in and out is difficult on a smartwatch touchscreen. Moreover, scrolling finger up and down can cause screen blockage. In this paper, we propose a new interactive technique with the smartwatches called TouchBand. It is a pressure-sensitive wristband that functions as an input tool for smartwatches’ scrolling and zooming tasks. Such input method will provide boarder interaction space and less occlusion to the display screen. To illustrate the potential of our approach, we describe the overall architecture of our system and present the applications that we used to illustrate how the apparatus work. In a controlled study with 12 participants, we found that TouchBand [insert results here].

## Author Keywords

Smartwatch; wristband; touch input; wearable.

## ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces – *input devices and strategies*.

# Introduction

Smartwatches have been gaining popularity over past few years because they support most smartphone functions, easy to wear and fashionable. It is anticipated that in the near future all smartphone functions will be on smartwatches, including calling. Because of this worldwide game-changer, many companies are entering the race of manufacturing more advanced smartwatches. However, smartwatches are not without constraints. Because the main input methods for traditional smartwatches on the market are display touch screens and physical buttons, it is difficult to carry out complex tasks on smartwatches due to the limited number of physical buttons and the inadequate size of the touch surface. The small screen can be partially or completely blocked by the user's finger when they interact with the device [10]. Furthermore, zooming in and zooming out on the screen can be inaccurate due to the size of the user's finger in relation to the size of the touchscreen. Our research aims to enhance users’ experience with smartwatches and increase efficiency and accuracy of scrolling and zooming through extending the touch surface to the wristband.

We propose placing Arduino touch sensors around the whole smartwatch wristband and using these sensors to identify scrolling and zooming input [needs update]. To scroll up and down, a user can move their finger along the long edge of the band. For long scrolls, a user can do multiple strokes, as they would on a normal touchscreen. We will not make the wristband identify horizontal scrolls because while the touch interaction space is bigger vertically (i.e. along the wristband), usually it is not bigger horizontally, since a lot of wristbands are more narrow than the touchscreen. To zoom, a user places two finger on the back of the wristband. To zoom in, the two fingers must move into opposite directions away from the bottom of the band and towards the screen. To zoom out, the two finger moves towards each other and towards the bottom of wristband [zooming needs update]. Our research question is: Is using the whole smartwatch wristband as touch input, including the back of the band, more convenient to users and will it make selection and zooming function faster and more accurate compared to the conventional touchscreen interaction?

We hypothesize that our input method can deliver faster zooming and scrolling speed (for lists) compared to conventional input method for smartwatches (i.e. display touchscreen) because in our design the finger will be moving along bigger interaction area. Moreover, moving the touch interaction off the screen will allow the user to not block the display screen while interacting with the device.

## Relevant Work

While no other input methods beyond smartwatch touchscreens and physical buttons have been implemented commercially, there are several studies that researched the possibility of extending the interaction surfaces beyond these two methods (such as using an air-magnet-pen or hand gestures). There are a few studies that used the wristband as an input method, but they either focused on developing the strap only for text entry or the wristband was only sensitive on the edges where the wristband touches the watch. We are studying the possibility of making the entire wristband sensitive so we can use the whole wristband as an input surface for scrolling and zooming. This will give the user the potential to perform more complex tasks with higher accuracy.

### External physical input tools

By using a magnetically driven input technique, Abracadabra [4] and Nenya [2] provided an unpowered wireless method for its users to interact with small mobile devices. It made use of magnetic sensing to expand the input area above the device and detect the finger’s movement along a one-dimensional polar and a 2D positional plane. However, this technique required the user to wear a magnet on the tip of their finger in Abracadabra and a ring on their finger in Nenya which is an additional component the user must manage. Moreover, Nenya was only able to interpret a set of eight commands.

### Built-in input methods

Perrault et al. [8] presented WatchIt, which uses watch’s wristband as interaction surface for eye-free command selection and execution. The prototype has two resistive sensors on each side of the band. This device uses two simple gestures for interaction: a pointing gesture for selecting an item in a list and sliding gesture consisting of sliding the fingertip along a half-band. Though a user study showed that the prototype was useful in silencing a vibrating ringer in meetings, WatchIt only supports interaction with one finger per band side.

Lyons et al. [7] presented Facet, a multi-display wrist-worn system consisting of independent touch-sensitive segments joined into a bracelet. Two common forms of interaction used on the multi-segment touchscreen were pinch and rotate. Using each segment’s accelerometer and magnetometer, they extracted orientation like pitch, roll, and yaw for each screen with respect to a common coordinate system. Since they did not conduct a user study, there was no proof of the device’s effectiveness.

Ahn et al. [1] explored pressure-sensitive multi-touch interactions with a smartwatch wristband. Their device had pressure-sensitive touch sensors on the wristband on either side of the smartwatch screen (but not the back). The sensors could detect tapping and flicking motions, as well as pressure input on part of the band that could be used as directional input. The device could also interpret flicking up and down motions as commands for copying and pasting. They demonstrated that this kind of device could be useful for many different tasks, although they did not conduct a user study to test the effectiveness of this device.

Funk et al. [3] experimented with two text entry methods on a touch-sensitive smartwatch wristband: sliding and multi-tapping. Both text entry layouts were a vertical arrangement of letters positioned to the sides of the watch display (again, nothing on the back of the wristband). The first had a narrow key for each character, and the user could slide their finger until the correct character was selected. The second had 3 letters per key, which were selected by multi-tap. They concluded that the multi-tap layout was faster in terms of words per minute, and was preferred by the study participants too. This project only supports text entry and no other task can be done with it.

### Skin as input tool

Knibbe et al. [6] proposed the use of hand gestures on the back of the smartwatch wearer’s hand as an input method. The proposed device supports a range of bimanual gestures that translate into commands for the smartwatch. The prototype built for this experiment had infrared proximity sensors on the sides of the watch and wristband. These sensors were used to identify different dynamic gestures from the watch hand, and to recognize bimanual gestures that were made by the other hand on the back of the watch hand. There were also piezoelectric sensors positioned underneath the watch to detect taps when the watch hand moved, and if there were actions at the back of the watch hand by the other hand. However, no user study was conducted to evaluate the prototype.

iSkin [11] is a skin-worn sensor touch input interface that can be used with smartwatches. Made of biocompatible materials, the prototype is thin, flexible and stretchable. It integrates capacitive and resistive touch sensing to detect touch input with two levels of pressure to account for stretching and bending. Furthermore, iSkin supports single or multiple touch areas of custom shape and arrangement, as well as more complex widgets, such as sliders and click wheels. The issue with this input tool is that when it stretches and deforms, the sensors resistance changes and it gets affected with fast and strong body movements.

Skinput [5] is a wristband which can identify the location of finger taps on the arm and hand. It analyzes mechanical vibrations that propagate through the body using an array of bio-acoustic sensors attached on a wristband. However, the setting for acoustic sensors should be changed based on the user sex, age and body mass index (BMI) because acoustic devices are affected by the wearer’s body composition.

Tomo [13] is a band worn around the wrist or the arm and it can identify the interior impedance geometry of the wearer’s arm by measuring the cross-sectional impedances between all pairs of eight electrodes resting on the user’s skin. The system can monitor and classify gestures in real-time. A user study concluded that Tomo could successfully identify hand and finger gestures with accuracy between 81% and 97%. However, like most other bio-sensing systems, results degrade when the system is re-worn at a later time, or worn by other users.

Rekimoto’s GestureWrist [9] is similar to the Tomo project. It uses higher-power capacitive and acceleration sensors to detect arm-shape changed based on. The work was at early stage when the paper was published and no further development was recorded.

# Methodology

## Apparatus

For the sensitive wristband we are fabricating, we used analog capacitive sensors attached to an Arduino on one end and to the wristband on the other. We used copper sheets as the capacitors. The Arduino sends the touch signals wirelessly to the smartwatch. The smartwatch we used had Andriod OS and the screen was 4”X4” [needs more details]

## Participants [to be updated]

We recruit 12 volunteers as participants. There were 6 males and 6 females in the age range of 20 to 30. The participants were a mixture of graduate and undergraduate students and were our colleagues in studying. While all participants were experts in using smartphones and laptops, around 40% of them had previous experience with smartwatches while the rest had not used a smartwatch before.

## Experimental Design

This experiment had two independent variables: input method and Task. The input method factor had two levels: the smartwatch display touchscreen and the touch-sensitive wristband. The task factor had two levels: scrolling and zooming. Speed and accuracy were the two dependent variables we were measuring. In every trial, a participant used one of the input tools (touchscreen or wristband) and attempt to finish one of the functions (scrolling or zooming). In each trial, we measured the speed and accuracy of completing each task.

The experiment had a within-subjects design, i.e. all participant will experience all levels in all factors. This design made the process much more streamlined and less resource heavy, since we had a limited number of participants and we wanted to collect as much data as we can for all factors and all levels. Moreover, by trying all input methods and tasks, participants could compare the input methods and express their preference which is as important to this experiment as the participants’ performance with each input method. Finally, a within-subjects design allowed us to reduce the amount of error arising from natural variance between individuals.

Each participant did 4 trials in the following orders to offset order effects (i.e. counterbalancing) if applicable (Table 1).

## Tasks and procedures

Each experiment lasted around 30 minutes per participant and was divided into four steps.

### Step 1: Introduction

We introduced to the participant verbally the experiment and how it would be conducted. Afterwards, we gave them written instructions about what would happen in the experiment, the different stages, and all the tasks they will be required to do at each stage. Finally, every participant was asked to answer a small questionnaire about themselves and their experience with smartwatches.

### Step 2: Training

Each participant was trained on how to use our touch-wristband prototype (and the smartwatch display touchscreen if they did not know already). We attached the prototype to their write then they had 3 minutes to explore the tools on their own and adapt to them. To accommodate for learning curve, participants were given a practice trial by doing one scrolling task and one zooming task.

### Step 3: Trials

Each participant completed four trials. The order of trials is different between participants as illustrated in Table 1. We used one smartwatch that was equipped with a touch-sensitive wristband for all trials and all participants. As Table 1 shows, trials 1 & 2 and trials 3 & 4 were paired up in terms of input method used. Depending on which trial it is, the participant strictly used that input method. We observed the participants closely to make sure they did not use the other input method.

We built an app that participants used on the smartwatch to test the different input methods and tasks. Once the app was running, there was only one option: new session. Each session had two tasks: scrolling and zooming so the testing of an input device for two tasks is done in one session. Hence, each new session in the app will cover two consecutive trials: trials 1 & 2 is a session and trials 3 & 4 is another session. Diagram 1 summarizes how the testing app works. When a participant had the smartwatch on them, we ran the app and click ‘new session’ which generated an ID. All data collected from this session would be saved under this ID. Afterwards, the screen went to the main menu which had three options: scrolling, zooming, and finish. We click on the task that should be done in this trial.

If scrolling task was selected, the participant saw a screen asking them to select a particular item. Once the participant is ready, they clicked start. A long list of items appeared and the participant had to navigate up and down to find the desired item and tap on the wristband. The app placed the desired item randomly in the second half of the list. The first item on the list was highlighted (which meant it can be selected) and as the user scrolled up and down the highlighting moved to the different items on the list. If the participant taped on the wristband on the wrong item, the app recorded it as wrong attempt (and we counted how many times they clicked on wrong items before selecting the right one for accuracy analysis later) but the screen did not change until the participant selected the right item. The app recorded the time from when the participant pressed the start button until they clicked the desired item. Once the right item was selected, the app went back to the scrolling start screen asking to click on a different item. Each participant had to do 10 rounds. Once the participant finished the required number of rounds, the screen went back to the main menu and we selected the next task or the finish option if this was the second task.

For the zooming task, the screen had a small black square and a participant had to zoom-in and zoom-out until the back square filled the screen. Once the participant zoomed enough, they tap on the wristband and it took them back to the zooming menu. On each test, we calculated the time from when the participant pressed the Start button until they taped on the screen or the wristband. Moreover, once the round is finished, we calculated the distance from the edges of square to the edges of the screen to measure accuracy. This was repeated 10 times. Once all rounds were done, the screen went back to the main menu.

When a participant finished the two tasks, we clicked on the ‘Finish’ button and the screen showed the ID of the session and we kept record of this ID in association with the participant for data analysis. There was a 2-minute break time until the next session, if there was one, which covered the two trials of the other input method.

### Step 4: Survey

Once a participant finished all four trials, they were asked to fill out a one-page survey regarding their experience with the two input methods. Questions include usage preference, adapting to the input method, and fatigue for both input methods for each task.

## Measures

We measured two variables:

1. Speed: for both tasks, speed is the time from when a participant pressed the Start button until they finished the round.

2. Accuracy:

* For scrolling task: (1) / (the number of times a participant clicks on an item in the list per round) [needs update]

For zooming task: (1) – (the distance from the edges of the current black square to the edges of the smartwatch / the distance from the edges of the original black square to the edges of the smartwatch) [needs update]

The above measures were crucial for determining whether our proposed input method was more suitable for scrolling through lists and zooming on smartwatches than the conventional one. By recording the time a user took to select an item on the screen or to navigate a list, we could compare the performance of both input methods. Moreover, we calculated the accuracy associated with each input method to assess the reliability of the input tool.

The survey at the end of the experiment was important to identify user experience with the different input methods. Even if the data from one input method showed significant superiority over the other input method, users’ personal experience with the device could show fatigue or hardship in learning or using a certain input method.

## Data collection

For each trial, we measured speed and accuracy. Our app had built-in code to time stamp whenever new screen showed up and calculated the time from the start of the task until a participant finished a round. The code also measured the accuracy by recording how many times an incorrect selection was made in the scrolling task and the black square dimensions’ in comparison to the smartwatch screen size in the zooming task.

To collect data about user experience with the different input methods, we personally observed each participant while they did the tasks and wrote notes about how they interacted with the input tools. Moreover, each participant filled out a questionnaire before the experiment indicating their skills with smartwatches and completed a paper survey at the end of their experiment to demonstrate their experience with the two input methods for the different tasks.

# results

[to be added]

# Disucssion

[to be added]

# future work

As a platform with unique affordances, there is an opportunity to develop additional interaction techniques. For example, a “slide up” finger movement could summon a menu that is hidden at the bottom of a segment, and a “slide down” gesture could hide the menu. Moreover, a sensitive wristband can be useful for eye-free interactions. For example, a user can tab on wristband to ‘snooze’ an event reminder, or reject, or put on silent, a phone call.

Our current prototype only considers interaction with one finger on one side of the band. Adding sensors on both sides of the band can make interaction more intuitive. For example for zooming, a user can place two finger on the back of the wristband. To zoom in, the two fingers move into opposite directions away from the bottom of the band and towards the screen. To zoom out, the two finger moves towards each other and towards the bottom of wristband. Moreover, the prototype can be programmed to understand the movement of two fingers on one band side (upper or lower). Allowing for pseudo-multitouch interaction would thus make the interaction with smartwatches richer. Furthermore, the screen can be used in conjunction with the smartwatch screen to make a full 360 interact surface. For example for zooming, a user can use two finger to pinch (in or out) on the screen and extend to the wristband or vice versa.

Currently, the copper sensors connection wires are arrayed on the wristband edge side by side then linked to the Arduino. The connections are made of copper (similar to the sensors) and once touched they send signals as if their corresponding sensors are touched. This can affect the input because a user may mistakenly press on these connection when they move their finger along the band. For future development, we will cover this portion completely in order to prevent corrupting the input because of faulty signals.

# conclusion

The display touchscreen as input method for smartwatches is limiting the amount of interaction with the device and can be easily blocked by a user finger. We proposed making a touch interface on the entire smartwatch wristband, including the back, to define interactions that can provide alternate methods of input scrolling and zooming. While previous works have attempted to solve similar issues, we believe that our approach addresses some of the pitfalls associated with previous works. By providing a touch interface at the back of wristbands, we expect that we will be able to define an input technique that does not occlude the screen while still providing an intuitive, integrated, and accurate method of navigating smartwatch interfaces. [to be updated with results]

# REFERENCES

1. Youngseok Ahn, Sungjae Hwang, HyunGook Yoon, Junghyeon Gim, and Jung-hee Ryu. 2015. BandSense: Pressure-sensitive multi-touch interaction on a wristband. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '15), 251-254. http://dl.acm.org/citation.cfm?doid=2702613.2725441
2. Daniel Ashbrook, Patrick Baudisch and Sean White. 2011. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 2043-2046. http://dl.acm.org/citation.cfm?doid=1978942.1979238
3. Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. 2014. Using a touch-sensitive wristband for text entry on smart watches. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14), 2305-2310. http://dl.acm.org/citation.cfm?doid=2559206.2581143
4. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 121-124. http://dl.acm.org/citation.cfm?doid=1622176.1622199
5. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 453-462. http://dl.acm.org/citation.cfm?doid=1753326.1753394
6. Jarrod Knibbe, Diego Martinez Plasencia, Christopher Bainbridge, Chee-Kin Chan, Jiawei Wu, Thomas Cable, Hassan Munir, and David Coyle. 2014. Extending interaction for smart watches: Enabling bimanual around device control. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14), 1891-1896. http://dl.acm.org/citation.cfm?doid=2559206.2581315
7. Kent Lyons, David Nguyen, Daniel Ashbrook, and Sean White. 2012. Facet: A multi-segment wrist worn system. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12), 123-130. http://dl.acm.org/citation.cfm?doid=2380116.2380134
8. Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. Watchit: Simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1451-1460. http://dl.acm.org/citation.cfm?doid=2470654.2466192
9. Jun Rekimoto. 2001. GestureWrist and GesturePad: unobtrusive wearable interaction devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers* (ISWC '01), 21-27.
10. Anne Roudaut, Stéphane Huot and Eric Lecolinet. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces* (AVI '08), 146-153. http://dl.acm.org/citation.cfm?doid=1385569.1385594
11. Martin Weigel1, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and J¨urgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). 2991-3000. http://dl.acm.org/citation.cfm?doid=2702123.2702391
12. Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost, Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15), 167-173. http://dl.acm.org/citation.cfm?doid=2807442.2807480