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

The popularity of smartwatches is on the rise due to their convenience and the various functions they offer. However, due to the small size of smartwatch's screen, there are a limited amount of touch gestures available for smartwatches and the screen can be easily blocked by a user's finger. In this paper, we propose TouchBand: a touch-sensitive wristband that functions as an input tool for the smartwatch's scrolling and zooming tasks. We describe the overall architecture of our system and present the application and apparatus that we used for testing our hypothesis. In a controlled study with 8 participants, we found that TouchBand performed almost as well as the touchscreen input method in terms of accuracy for both scrolling and zooming task. However, the task completion time for both tasks for our prototype was almost double to four times that of using the touchscreen. Because ANOVA analysis showed no statistical significance, we believe that with a better touch sensor and direct data transfer, our prototype could perform as well as, or even better than, a touchscreen.

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 and are easy to wear and fashionable. It is anticipated that in the near future all smartphone functions will also be on smartwatches, including calling [8]. Because of this, many companies worldwide are entering the race of manufacturing more advanced smartwatches. However, 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 [11]. 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 the efficiency and accuracy of scrolling and zooming through extending the touch surface to the wristband.

Figure 1. By using TouchBand, a user will not block the screen when interacting with the smartwatch. A user can slide their finger along the band to scroll and zoom.

This study involves placing capacitive touch sensors on the smartwatch wristband and using these sensors to recognize scrolling and zooming gestures (Figure 1). 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. In this study, we did 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 wider horizontally. To zoom on the wristband, a user places one finger on the back of the wristband. To zoom in, the user can scroll down, and to zoom out, the user scrolls up. 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 the display touchscreen for smartwatches because in our design the finger is moving along a 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.

Related 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. [9] 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 the 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 also preferred by the study participants. However, this device 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 [12] 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 by 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 settings for the acoustic sensors have to be changed based on the user's 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 [10] 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 an early stage when the paper was published and no further development was recorded.

TOUCHBAND DESIGN CONCEPT

We designed a touch sensor that is placed on the side of the wristband, extending from the edge of the screen to the back of the band. For practical reasons we only placed sensors on one side, although sensors could easily be added to the other side to cover the whole wristband.

This touch-sensor enabled wristband, which we refer to as the TouchBand, is used as a touch input surface instead of the display screen, which allows us to use the existing area of the watchband as a larger touch surface. The TouchBand contains 6 sensors placed in a row, so that touch movements can be detected as the user slides their finger along the sensors.

For these experiments, motion is only detected along one axis (the length of the wristband), and only single touches are considered. We did not include multi-touch gestures such as pinch-to-zoom because of time constraints.

METHODOLOGY

Apparatus

For the touch-sensitive wristband we used analog self-capacitive sensors made of copper sheets attached to one side of the smartwatch wristband, with the signals processed by an Arduino board (See Figure 2). The Arduino sends the touch signals wirelessly to a web server, which forwards the information to an app on the smartwatch. The smartwatch we used for this study is a Sony Smartwatch 3 running Android OS 6.1 and the screen size is 1.4"X1.4".

Participants

We recruited 8 volunteers as participants. There were 3 males and 5 females in the age range of 20 to 30. The participants were a mixture of graduate and undergraduate students and were our colleagues at the university. While all participants were experts in using smartphones and laptops, none of them had extensive previous experience with smartwatches.

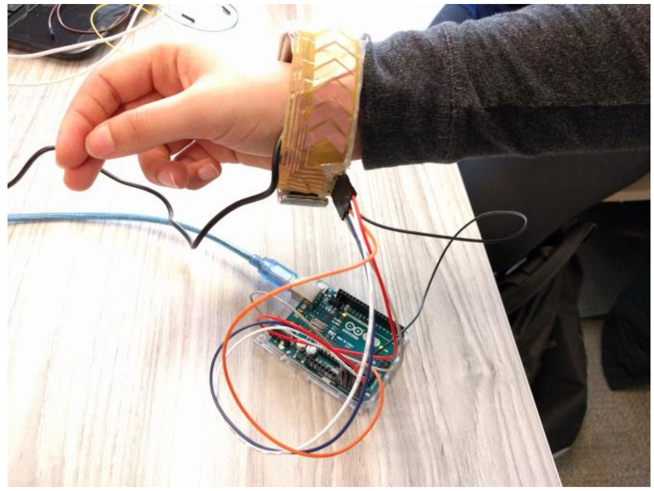

Figure 2. Hardware apparatus. There are six copper sensors on the wristband. Wires connect the sensors to the Arduino which is connected to a laptop to receive the touch data.

Experimental Design

The experiment was a within-subjects design, i.e. all participants completed 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 possible 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.

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. The two dependent variables were speed and accuracy. In every trial, a participant used one of the input methods (touchscreen or wristband) and attempted to complete one of the tasks (scrolling or zooming) multiple times. We measured the speed and accuracy for each trial.

The participants did 4 blocks of 5 trials in different orders to offset order effects (i.e. counterbalancing).

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 to 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. Every participant was also 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 TouchBand prototype (and the smartwatch display touchscreen if they did not know already). We allowed each user to practice using the wristband before the actual tasks began.

Step 3: Trials

Each participant completed 4 blocks of 5 trials. We used the same smartwatch and touch-sensitive wristband for all trials and all participants. Blocks 1 & 2 and blocks 3 & 4 were paired up in terms of input method used. Depending on which trial it was, the participant strictly used that input method. For the touchscreen trials, we disabled the touch sensor on the wristband to prevent accidental touch input.

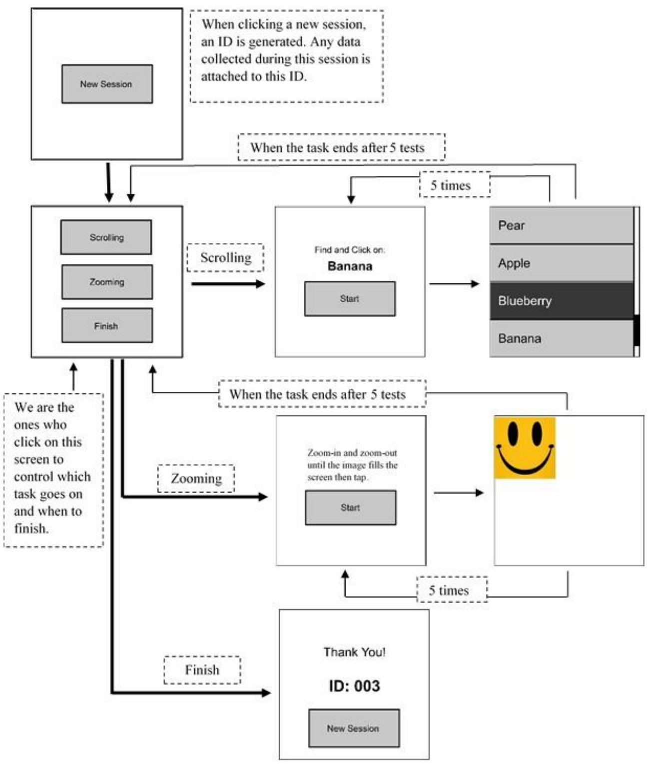

Figure 3: Testing App format

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 covered two consecutive trial blocks: blocks 1 & 2 are a session and blocks 3 & 4 are another session. Figure 3 summarizes how the testing app works. When a participant had the smartwatch on them, we ran the app and clicked 'new session' which generated an ID. All data collected from this session was saved under this ID. Afterwards, the screen went to the main menu which had three options: scrolling, zooming, and finish. We selected 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 was ready, they clicked start. As Figure 4 illustrates, 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 visible item on the list was highlighted (which meant it could be selected) and as the user scrolled up and down the highlighting moved to the different items on the list. If the participant tapped on the wristband on the wrong item, the app recorded it as a 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 5 rounds with each input method. 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.

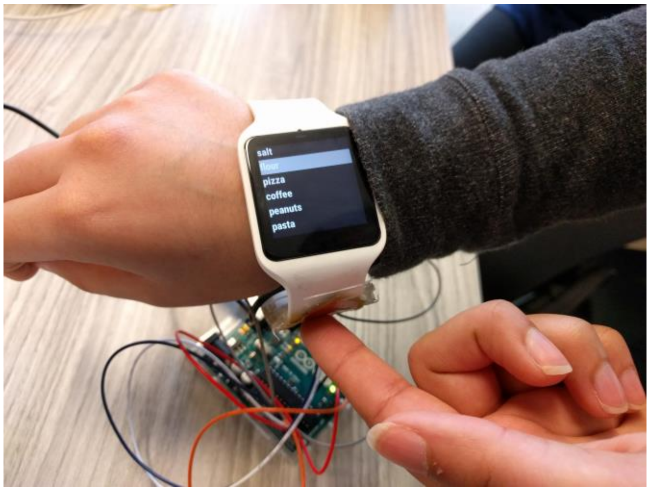

Figure 4. For the scrolling task, a user navigates a list then taps on the item when finished.

For the zooming task, the screen had a square picture of a smiley face and the participant had to zoom-in and zoom-out until the picture filled the screen as Figure 5 shows. Once the participant zoomed enough, they tap on the wristband and it took them back to the zooming menu. This was repeated 5 times. For each round, we calculated the time from when the participant pressed the Start button until they tapped on the screen or the wristband. We calculated the ratio of the re-sized image to the expected image size (the size of the screen) and recorded this as the accuracy metric. Once all rounds were done, the screen went back to the main menu.

Figure 5. For the zooming task, a user zooms in and out until the image fills the screen then taps to finish

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: speed and accuracy. For both tasks, speed is the time from when a participant pressed the start button until they tapped on the band or screen to finish the trial.

For the scrolling task, accuracy is the number of correct selections out of the total number of selections, i.e. 1 / # of taps, since the task finishes once the participant taps on the correct item.

For the zooming task, participants were asked to re-size the image until it matched the size of the screen, so accuracy is the ratio of the re-sized image to the screen size (the absolute value of the difference in size / the screen size).

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 app automatically measured the time and accuracy for each task and recorded it in a table.

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

	Scrolling Task	Zooming Task	*Mean*
Touchscreen	16.6	5.1	10.9
TouchBand	72.6	11.3	41.9
Mean	44.6	8.2	26.4

Table 1. Average completion time (s). There was a significant difference in the completion time with the TouchBand falling behind the touchscreen for both tasks.

The results by device and task for time completion are shown in Table 1. The grand mean for task completion time was 26.4 seconds. The touchscreen was faster at 10.8 seconds, while the sensitive wristband was slower at 41.9 seconds. The main effect of device on task completion time was statistically not significant ($F_{1,14} = 4.119$, ns). The task effect was significant, however. Task completion time was 44.6 seconds for the scrolling task. The zooming task was enormously faster at 8.2 second. the difference was statistically significant ($F_{1,7} = 7.497$, $p < 0.05$). There was no significant Device X Task interaction effect ($F_{1,14} = 3.104$, ns).

	Scrolling Task	Zooming Task	*Mean*
Touchscreen	0.79	0.91	0.85
Touch-band	0.86	0.80	0.74
Mean	0.74	0.86	0.80

Table 2. Average task accuracy. The accuracy rates were comparable between the TouchBand and the touchscreen for both tasks.

The results by device and task for accuracy are shown in Table 2. The grand mean for task accuracy was 80%. The touchscreen accuracy was higher at 85%, while the wristband as input method had 74% accuracy rate. The main effect of device on task accuracy was not statistically significant ($F_{1,14} = 4.742$, $p > 0.05$). The task effect was also not statistically significant. The accuracy was 73% for the scrolling task and 86% for the zooming task. The difference was not statistically significant ($F_{1,7} = 3.226$, $p > 0.05$). Also, there was no significant difference in the Device X Task interaction effect ($F_{1,14} = 0.007$, ns).

DISCUSSION

When asked on the questionnaire how difficult it was to learn to use the wristband, the median response was 2 on a scale from 1 (easy) to 5 (difficult). When asked how intuitive using the wristband was for the scrolling and zooming tasks, the median response for both was 3 on a scale from 1 to 5.

Most, but not all users expressed a preference for the touchscreen. One participant said they preferred the TouchBand for scrolling, and 2 said they preferred the TouchBand for zooming. In addition, 2 participants said that using the touchscreen caused fatigue, and 3 participants said that using the TouchBand caused fatigue, so both input methods were somewhat taxing to the user.

Even though the touchscreen input method performed better than the TouchBand in every case, the difference is not statistically significant. With a better touch sensor and direct data transfer, a TouchBand could be just as good as a touchscreen.

The main issue with the TouchBand prototype seems to be the low quality of the sensor used, which was made from thin sheets of copper, and the delay in sending the touch information over the network. In addition, the touch input surfaces are not truly comparable. The TouchBand consists of 6 distinct touch sensors, and only detects one axis of motion. The TouchBand sensor would be a better comparison to the touchscreen if it was a continuous surface with x and y coordinates. A continuous sensor would allow for more fine-grained input and a smoother experience. The touchscreen also sends input signals directly to the Android system, while the TouchBand sends the information over the network, which results in a delay in processing the touch information.

With better hardware and a direct connection to the watch, the gestures could be interpreted much quicker, resulting in a better user experience. A touch sensor with two axes would be more similar to the typical touchscreen experience, and allow for more input gestures.

FUTURE WORK

Currently, the connection wires for the touch sensor are arrayed on the wristband edge side by side then linked to the Arduino. The connections are made of copper, similar to the input sensors sheets. Once these links are touched, they send signals as if their corresponding sensors are touched. This can affect the input because a user may touch these connections by mistake when they move their finger along the band. For future development, we could cover this portion completely in order to prevent corrupting the input because of faulty signals.

The copper sensors we used to identify finger location on the wristband were poor in quality. For future development, we will most likely change them into more durable sensors or at least seal them with a silicon cover to prevent sensors from wearing off after multiple usages.

There is an opportunity to develop additional interaction techniques with the wristband. 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. A touch-sensitive wristband could also be useful for eyes-free interactions. For example, a user could tap on wristband to snooze an event reminder, or to reject or silence 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 could make interaction more intuitive. For example, for zooming a user could place two fingers 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 fingers move towards each other and towards the bottom of wristband. Moreover, the prototype could be programmed to understand the movement of two fingers on one band side (upper or lower). Allowing for pseudo-multitouch interaction would provide a richer interaction with the smartwatch. Furthermore, the touch sensor could be used in conjunction with the smartwatch screen to make a full 360 degrees interaction surface. For zooming, a user could use two fingers to pinch (in or out) on the screen and extend to the wristband or vice versa.

CONCLUSION

Using only the display touchscreen as the touch input for smartwatches limits the amount of interaction with the device and the screen can be easily blocked by a user’s finger. We proposed making the entire smartwatch wristband touch sensitive, including the back, to define interactions that can provide alternate methods of scrolling and zooming. While this prototype was generally slower than the touchscreen, the difference in speed and accuracy was not statistically significant, even though we were using a cheap sensor. With a commercial grade sensor and fully developed software, a touch-sensitive wristband has the potential to be a viable extension of smartwatch interaction. 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.

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. Tim Moynihan. 2015. Samsung's Slick New Smartwatch Makes Calls Without a Phone. In *Wired*. https://www.wired.com/2015/08/samsungs-round-smartwatch-official-makes-calls
9. 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
10. 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.
11. 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
12. 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
13. 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