WatchMI: Pressure Touch, Twist and Pan Gesture Input on Unmodified Smartwatches

Hui-Shyong Yeo [1]
hsy@st-andrews.ac.uk

Juyoung Lee [2]
ejuyoung@kaist.ac.kr

Andrea Bianchi [3]
andrea@kaist.ac.kr

Aaron Quigley [1]
aquigley@acm.org

[1]School of Computer Science, University of St Andrews, Scotland, United Kingdom
[2]Graduate School of Culture Technology, KAIST, Republic of Korea
[3]Department of Industrial Design, KAIST, Republic of Korea

Figure 1. The three main techniques proposed in our paper: (a) Omni-direction pressure touch (b) Bi-direction twist and (c) Omni-direction panning.

ABSTRACT

The screen size of a smartwatch provides limited space to enable expressive multi-touch input, resulting in a markedly difficult and limited experience. We present *WatchMI: Watch Movement Input* that enhances touch interaction on a smartwatch to support continuous pressure touch, twist, pan gestures and their combinations. Our novel approach relies on software that analyzes, in real-time, the data from a built-in Inertial Measurement Unit (IMU) in order to determine with great accuracy and different levels of granularity the actions performed by the user, without requiring additional hardware or modification of the watch. We report the results of an evaluation with the system, and demonstrate that the three proposed input interfaces are accurate, noise-resistant, easy to use and can be deployed on a variety of smartwatches. We then showcase the potential of this work with seven different applications including, map navigation, an alarm clock, a music player, pan gesture recognition, text entry, file explorer and controlling remote devices or a game character.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User interfaces - Input devices and strategies.

Author Keywords

Rich Touch; Smart watch; Small screen; Wearable devices.

INTRODUCTION

Arm and wrist-worn devices such as fitness trackers or smartwatches are seeing growing adoption. Many of these devices have screen displays for interaction, such as the Samsung Gear [17], Fitbit Surge [5] or Apple Watch [2]. However, due to the average width of human fingers [10] and the limited screen size or interaction surface, interacting with such devices can be difficult and cumbersome. Existing challenges with interaction on mobile devices, such as the "fat finger" problem [3, 18] or occlusion [22] can be exacerbated with small interaction surfaces on such devices. Further challenges now present themselves, such as how to support the range of multi-touch interactions which consumers commonly expect. Researchers and device manufacturers have attempted to address many of these challenges by incorporating new input modalities, such as pressure sensitive touch [2, 16], bezel rotation [17], shear gestures [6, 8] and a combination of these [14, 23]. Such approaches, however, require additional hardware and moving parts which can increase both cost and weight of the devices, limiting their adoption for common use.

In this paper, we propose a new approach to sense continuous rate-based "touch pressure", "twist angle" , "pan movement" and the combination of these on unmodified smartwatches. Our goal is to expand the input expressiveness, so that users can interact with small devices without suffering the limitations of the small screen size. Our techniques leverage the built-in IMU already available in almost every smartwatch and smart wristband in the consumer market, and does not require additional sensor or hardware modification of the watch. The software utilizes a multi-modal approach combining touch events with the orientation of the watch to enrich the input capabilities, while also effectively rejecting false positives. In the following sections we introduce related work, a description

of our system, an evaluation to gauge its performance, and illustrate limitations and future avenues of research.

RELATED WORK

Prior watch-based interactions have extensively relied on custom hardware extensions or modifications to achieve the input expressiveness that is usually not possible on off-the-shelf devices. WatchIt [15] and BandSense [1], for example, expands the input area to the wristband using custom pressure sensors. Alternatively, Xiao et al. [23] and SkinWatch [14] enhance the interaction by supporting tilt, twist and pan on the watchface with Hall-effect joysticks or photo-reflective sensors. Additional sensors can also extend the interaction space around the watch. In EdgeSense [12] input is expanded to the side of the watch by using an array of capacitive sensors. SkinButtons [9] and Lim et al. [11] have expanded the input area to the skin surface beside the watch, using infrared sensors, while zSense [20] expands the input to mid-air area around the watch, using LEDs and infrared sensors. These approaches rely on additional hardware but result in a wide range of interactions that are not possible on commodity smartwatches.

An alternative approach is to instrument the user's finger which is used to interact with the smartwatch. For example, NanoStylus [22] presents a method to include a finger-mounted stylus to improve touch precision when interacting on smartwatches. In Abracadabra [7] the authors expand the input to a mid-air area around the watch but requires a user to wear a magnet ring. By contrast, FingerPad [4] proposes an interaction method using different sides of the finger but it requires augmenting the finger with a magnet and Hall-effect sensor grid.

Finally, there are also examples of interaction techniques for smartwatches which solely rely on the existing hardware and sensors available on off-the-shelf wearable devices. Clicking, covering, holding, tapping and swiping are some examples of the interaction techniques available within commodity devices today. FingerPose [24] and Finger Orientation [19] attempt to infer finger angle on a touch screen while Beats [13] expands interaction by proposing temporal touches. Similarly to this last group of interfaces, our system does not require any additional hardware sensors, but it also allows highly expressive input by supporting continuous and different levels of touch pressure, pan movement and twist angle using only the data collected from the built-in IMU sensor.

PROTOTYPE AND INTERACTION

We prototype our system using the Android Wear SDK for an unmodified LG Urbane smartwatch, which mounts an IMU with sensor fusion composed by an Invensense M651 6-axis accelerometer and gyroscope, and by an Asahi Kasei AK8963 3-axis compass sensor. Our system adds omni-direction pressure touch, bi-direction twist and omni-direction pan that, in a user's eyes, operate like many familiar devices, such as music keyboards (pressure touch), volume knobs (twist) or joysticks (panning). The basic working principle is simple. The human skin is elastic and stretchable, so are typical watch straps made from leather, rubber or metal meshes. These material properties allow us to measure the changes in a watch's tilt angle due to skin and strap deformation when pressed or twisted with force. Using the IMU tilting vector data, we can then estimate the amount of pressure and the direction of the force applied. When the force is released, the elasticity of the skin and strap naturally pushes the watch face back to its original position.

Our software collects in real-time the rotation vector (the orientation measure of the device based on the integration of the internal accelerometer, gyroscope and magnetometer readings) from the IMU sensor, at a sampling frequency of about 100 Hz. It gives the 3-axis orientation of the watch as yaw, pitch and roll angles. Since we collected data using the Android sensor fusion, which is factory calibrated and reliable against noise, no further calibration was necessary. When a touch event occurs (i.e., when the user touches the screen), our software records the current orientation as the initial value. Then it constantly measures the latest value and calculates the difference from the initial value as a delta vector. This delta of relative change is then normalized and linearly mapped to screen coordinates, with a minimum threshold and multiplier determined in pilot testing. For pressure touch and panning, we use only the pitch and roll value while in twisting we only use the yaw value. This simple calculation performed at every touch makes the system resistant to noise (e.g., unintentional arm movements) and allows users to comfortably input on the smartwatch, regardless of its initial orientation. Finally, depending on the technique being used, we visualize the delta as a growing bar (pressure), rotating arc (twist) and a cursor (panning) that moves in a 2D plane, as shown in Figure 1 and Figure 2.

Since the IMU senses any changes in acceleration or orientation, which can be caused by any hand motion, we use the touch event as a trigger, and stop measuring the value when the user lifts the fingers from the screen. This is simple but very effective: if no touch is detected, the changes in IMU values are discarded, preventing us interpreting accidental motions such as hand waving or gesturing. In addition, we can turn off IMU sensing when it is not necessary, thus reducing battery consumption. A minimum threshold also clearly differentiates between a normal touch or the enhanced touch (pressure, twist or pan). Finally in the pressure touch mode, we use a hybrid approach checking both the touch location and the tilt direction in order to ignore the accidental triggering of events (e.g., if the touch position does not match the tilt direction).

EVALUATION STUDY

We designed a user study to assess the usability of the three bespoke input interfaces - pressure, twist and pan input. To characterize the performance of each condition, we modified the three interfaces and collected measures for input entry times and errors (Figure 2). The input for each interface was discretized in 24 selectable regions. In the twist interface, 24 identical regions each spanning 15° were radially placed around the edge of the screen. In the pressure and pan interfaces, the screen was divided into 8 regions, each spanning 45° and subdivided in 3 levels along the radius for different strengths applied. There is a minimum threshold before entering the first level. A visual cursor on the screen allows a user to navigate across the 24 regions. To select a target region the user needs to remain within the specific region for the duration

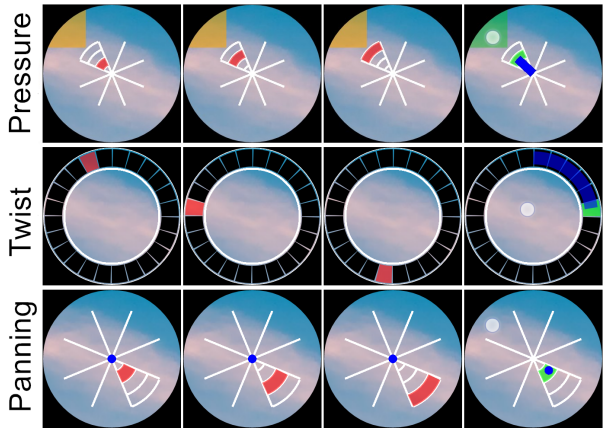

Figure 2. Experiment tasks, from top: 3 levels of pressure targets along the radius (in 8 directions), middle: 3 levels of twist (nearest 4, medium 4, farthest 4 targets in both directions) and bottom: 3 levels of panning targets along the radius (in 8 directions).

of 1000ms. The 1000ms duration is restarted if the cursor exits the target region accidentally.

We recruited 12 volunteers (3 female, 3 left-handed) from a local university, including undergraduate and graduate students, researchers and staff, aged between 20 and 36 (M: 25.8, SD: 5.2). Five reported familiarity with wearable devices, and of these, four said they own a smart wristband or fitness tracker. Participants were compensated with 5 GBP.

The experiment was conducted in a quiet lab and took between 40 to 60 minutes to complete. Before the study we debriefed participants and collected demographics. The study followed a repeated-measures design, and the three interface conditions were balanced in a Latin-squared design order. Each condition was identical. Participants received a demonstration of one of the interfaces and had a chance to practice with it for about 2 minutes. They then were required to complete a targeting task - each of the 24 regions were highlighted in randomized order and a trial consisted of selecting the correct region. A successful trial resulted in prompting the next randomized region, while a failure would require participants to try the selection again. Each condition required 144 correct trials (24 regions x 6 repetitions), of which the first 48 trials were considered practice and discarded in the analysis. During the targeting tasks, users remain seated on a chair and the arm with the smartwatch rested on the table. After completing the targeting task, participants rated the perceived usefulness and usability of each interface using a 7-points Likert scale. After each condition, participants were interviewed and asked to comment about each of the interfaces and their possible applications. For each of the three interfaces we collected 1152 valid trials (12 participants x 24 regions x 4 repetitions) for the analysis.

Results

The analysis is based on one-way ANOVA tests followed by post-hoc analysis using Bonferroni correction, with α=0.05. Mauchley's test assessed sphericity, and, if violated, Greenhouse-Geisser corrections were employed. Pressure

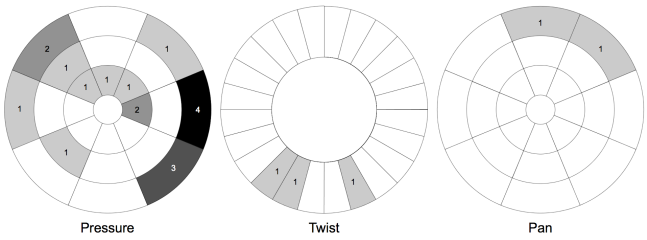

Figure 3. Error occurrence by location for three techniques. Errors for left-handed people were vertically mirrored so to mimic right-handed users' input.

was the slowest input interface (M: 2.6s, SD: 0.9) followed by Twist (M: 2.5s, SD: 0.6) and then Pan (M: 2.1s, SD: 0.6). Time differences are statistically significant ($F(1.84, 2120.9)=104.1$, $p<0.01$, $\eta_p^2=0.08$) with Bonferroni post-hoc pairwise comparisons significant as well ($p<0.01$). While the overall input entry time took about 2.4 seconds, it is interesting to note that on average only 62% (SD: 0.07) of this time was used for the actual input (1.5s including 1s dwell time) while the remainder of the time was devoted to navigate to the next task and decipher the instructions for the task. In terms of errors, Pressure unsurprisingly performed worse than the other two interfaces with a 1.6% error rate (in total, 18 failed trials), compared with the 0.3% (3 failed trials) and 0.2% (2 failed trials) of Twist and Pan. Differences were again statistically significant ($F(2, 2302)=11$, $p<0.01$, $\eta_p^2=0.01$) and the post-hoc tests corroborated that Pressure was worse than the other two ($p<0.01$). Figure 3 shows in detail the number of errors for the 24 selectable regions. For Twist and Pan all reported errors consisted of selecting the region immediately before or after the target. For pressure, the average distance between the erroneous selection and the target was of 1.9 (SD: 2.1) regions. Finally, we used a Chi-square test to analyze the Likert scores and found a statistically significant difference in perceived usefulness ($\chi 2$ (2, N=12) = 7.03, p = 0.03) and usability ($\chi 2$ (2, N=12) = 8.1, p = 0.017) of the three prototypes, with Twist reported to be most useful (M: 6.5, SD: 0.7) and usable (M: 6, SD: 0.7), followed by Pan (usefulness = 5.8 SD: 1; usability = 5.7 SD: 1.2) and Pressure (usefulness = 5.7 SD: 1; usability = 4.3 SD: 1.1).

Discussion and Findings

Overall, participants were able to complete all input tasks in less than 2.4 seconds (including the 1 second dwell time) and with an average error rate of 0.7%. During the interviews, most participants described the system as "responsive" and "easy to learn", findings which were numerically corroborated in the post-experiment questionnaire. The Pressure interface was described to be the most challenging of the three, as it was difficult to apply the correct force in the correct direction. Moreover, 55% of errors (10 out of 18 errors) for the Pressure interface were performed by only two of the twelve participants. Analyzing the video recordings of the experiment, we observed that P1 and P11 strapped the watch tightly on the wrist and in proximity of the *Ulnar styloid process* (the extruding part of the Ulnar bone near the wrist), so that the watch could barely register any motion when the screen was pressed,

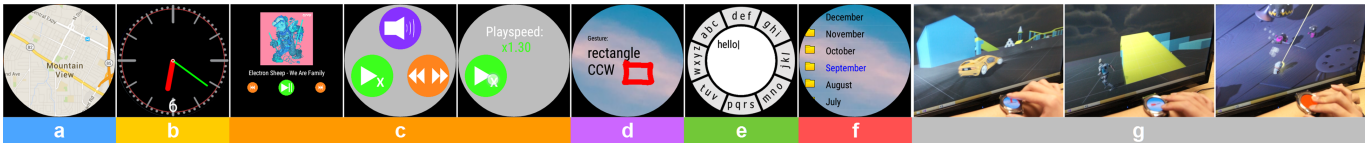

Figure 4. Screenshots of the applications. From left to right: (a) map, (b) alarm clock, (c) music player, (d) gesture recognizer, (e) text entry, (f) file explorer, and (g) controlling remote device and game character.

causing errors on the right hemisphere. Though further investigations are required, we suspect that the way the watch is strapped on the wrist might impact on the overall recognition performance of the system. Finally, the errors occurred on the left hemisphere are mainly due to occlusion, as the right hand clicking on the left portion of the watch screen (strapped on the left wrist) occludes the view from the user line of sight.

During the interview several participants noted that the Twist interface in practice requires the usage of three fingers - two fingers to hold the rim of the watch, and one finger to act as a pivot while touching the screen. Users noted that, not only is such a gesture complex, but it also results in large portions of the screen being occluded. Ways to overcome this problems were suggested and employed by the users. For example, it is possible to place the pivoting finger on the corner of the screen. Users also suggest to use an alternative visualization for the Twist interface. We note however that the visualization we chose was only meant for use in the study and we share the participants concerns that in practice, Twist techniques should rely on simpler visuals so as to avoid cluttering the screen.

APPLICATIONS

To illustrate the potential and immediate feasibility of our approach, we developed 7 applications (Figure 4 and Figure 5) to showcase these 3 interaction techniques - a) Continuous map navigation b) Alarm clock c) Music player d) Pan gesture recognition e) Text entry f) File explorer and g) Controlling remote devices. These demos take advantage of the continuous rate control based on the strength applied, such as continuously navigating and zooming the map (Figure 5), twisting the clock hand, adjust the volume or controlling game characters. In text entry, different pressures disambiguate between characters, similar to how MultiTap operates. In file explorer, panning up/down scroll through the list while panning left/right enter or exit the current folder. Finally, pan gesture recognition [21] acts as a shortcut to launch apps. Please refer to our supplement video figure to see how each demo works in practice.

Our techniques, instead of being used for new interactions, can be also use for reserved functions. For example, all Android smartwatches reserved the left and down swipe for OS functionality - exiting app and showing top menu bar. This severely limits the interaction capability. Our technique can free the reserved left and down swipe for more basic purposes.

LIMITATIONS AND FUTURE WORK

This work has a number of limitations but also opportunities for improvement in future revisions. First of all, our techniques require a touch event on the capacitive screen in order to register an event - in other words, the software requires the user to touch the screen in order to discriminate intentional input from accidental arm movements. This limitation has practical consequences: capacitive input is required so that the watch screen area is partially occluded during input interaction, especially during multi-touch selections (e.g., Twisting). Similarly, touching a smartwatch on its rim will, for most of currently available devices, does not trigger a touch event and the input will not be recognised by the system. Future work will attempt to address this problem by using accelerometer or other sensor data to replace the need for triggering a touch event, perhaps a conductive tape on the rim to route the touch.

Figure 5. Map application running on LG watch Urbane: Twist to zoom and pressure touch to move the map at different speed.

Another limitation of this work is that, though the three interaction techniques we presented work well independently as shown in our study, we did not collect evidence of how they work when used simultaneously. Our applications showcased that it is possible to combine and seamlessly integrate them, but future work will need to address in more detail about possible degraded performance or mis-recognitions of triggering events. Future work will also attempt to study these techniques when applied to wearable devices with different form factors or located on different body locations beyond the wrist (e.g., upper arm band, necklace, belt) or for eyes-free input interaction. Finally, we plan to measure the sensing accuracy and study how our techniques perform while a user is engaged in activities requiring motion, such as walking or running.

CONCLUSIONS

In this work, we designed and implemented three techniques for augmenting normal touch input on a smartwatch, which lead to many potential use cases. Our techniques use only the signal from integrated IMU to infer the action and its magnitude. Thus, it can already work on most of the smartwatch in the market, perhaps via a simple software update. Our user study showed that our techniques are immediately feasible, with users' input accuracies in excess of 98.4%.

ACKNOWLEDGEMENTS

We thank all volunteers, reviewers and Scottish Informatics and Computer Science Alliance (SICSA). Andrea Bianchi was supported by the MSIP, Korea, under the G-ITRC support program (IITP-2016-R6812-16-0001) supervised by the IITP.

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)*. ACM, New York, NY, USA, 251–254. DOI: http://dx.doi.org/10.1145/2702613.2725441
2. Apple. 2015. Apple Watch Force Touch. (2015). http://www.apple.com/watch.
3. Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 39–48. DOI:http://dx.doi.org/10.1145/2371574.2371582
4. Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 255–260. DOI: http://dx.doi.org/10.1145/2501988.2502016
5. Fitbit. 2016. Fitbit Surge. (2016). https://www.fitbit.com/uk/surge.
6. Chris Harrison and Scott Hudson. 2012. Using Shear As a Supplemental Two-dimensional Input Channel for Rich Touchscreen Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 3149–3152. DOI: http://dx.doi.org/10.1145/2207676.2208730
7. 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)*. ACM, New York, NY, USA, 121–124. DOI: http://dx.doi.org/10.1145/1622176.1622199
8. Seongkook Heo and Geehyuk Lee. 2011. Force Gestures: Augmenting Touch Screen Gestures with Normal and Tangential Forces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 621–626. DOI: http://dx.doi.org/10.1145/2047196.2047278
9. Gierad Laput, Robert Xiao, Xiang 'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin Buttons: Cheap, Small, Low-powered and Clickable Fixed-icon Laser Projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 389–394. DOI: http://dx.doi.org/10.1145/2642918.2647356
10. Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 669–678. DOI: http://dx.doi.org/10.1145/2702123.2702388
11. Soo-Chul Lim, Jungsoon Shin, Seung-Chan Kim, and Joonah Park. 2015. Expansion of Smartwatch Touch Interface from Touchscreen to Around Device Interface Using Infrared Line Image Sensors. *Sensors* 15, 7 (2015), 16642. DOI:http://dx.doi.org/10.3390/s150716642
12. Ian Oakley and Doyoung Lee. 2014. Interaction on the Edge: Offset Sensing for Small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 169–178. DOI: http://dx.doi.org/10.1145/2556288.2557138
13. Ian Oakley, DoYoung Lee, MD. Rasel Islam, and Augusto Esteves. 2015. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1237–1246. DOI: http://dx.doi.org/10.1145/2702123.2702226
14. Masa Ogata and Michita Imai. 2015. SkinWatch: Skin Gesture Interaction for Smart Watch. In *Proceedings of the 6th Augmented Human International Conference (AH '15)*. ACM, New York, NY, USA, 21–24. DOI: http://dx.doi.org/10.1145/2735711.2735830
15. 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)*. ACM, New York, NY, USA, 1451–1460. DOI: http://dx.doi.org/10.1145/2470654.2466192
16. Jun Rekimoto and Carsten Schwesig. 2006. PreSenseII: Bi-directional Touch and Pressure Sensing Interactions with Tactile Feedback. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1253–1258. DOI: http://dx.doi.org/10.1145/1125451.1125685
17. Samsung. 2015. Samsung Gear S2. (2015). http://www.samsung.com/global/galaxy/gear-s2/.
18. Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'05)*. Springer-Verlag, Berlin, Heidelberg, 267–280. DOI: http://dx.doi.org/10.1007/11555261_24

19. Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. 2009. Detecting and Leveraging Finger Orientation for Interaction with Direct-touch Surfaces. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 23–32. DOI: http://dx.doi.org/10.1145/1622176.1622182

20. Anusha Withana, Roshan Peiris, Nipuna Samarasekara, and Suranga Nanayakkara. 2015. zSense: Enabling Shallow Depth Gesture Recognition for Greater Input Expressivity on Smart Wearables. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3661–3670. DOI: http://dx.doi.org/10.1145/2702123.2702371

21. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/1294211.1294238

22. Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 447–456. DOI: http://dx.doi.org/10.1145/2807442.2807500

23. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 193–196. DOI:http://dx.doi.org/10.1145/2556288.2557017

24. Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. DOI: http://dx.doi.org/10.1145/2817721.2817737