WristWhirl: One-handed Continuous Smartwatch Input using Wrist Gestures

Jun Gong[1], Xing-Dong Yang[1], Pourang Irani[2]
Dartmouth College [1], University of Manitoba[2]
{jun.gong.gr; xing-dong.yang}@dartmouth.edu, pourang.irani@cs.umanitoba.ca

ABSTRACT
We propose and study a new input modality, WristWhirl, that uses the wrist as an always-available joystick to perform one-handed continuous input on smartwatches. We explore the influence of the wrist's bio-mechanical properties for performing gestures to interact with a smartwatch, both while standing still and walking. Through a user study, we examine the impact of performing 8 distinct gestures (4 directional marks, and 4 free-form shapes) on the stability of the watch surface. Participants were able to perform directional marks using the wrist as a joystick at an average rate of half a second and free-form shapes at an average rate of approximately 1.5secs. The free-form shapes could be recognized by a $1 gesture recognizer with an accuracy of 93.8% and by three human inspectors with an accuracy of 85%. From these results, we designed and implemented a proof-of-concept device by augmenting the watchband using an array of proximity sensors, which can be used to draw gestures with high quality. Finally, we demonstrate a number of scenarios that benefit from one-handed continuous input on smartwatches using WristWhirl.

Author Keywords
One-handed interaction; smartwatch; smartwatch input; continuous input; gestural input.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION
Interacting with a smartwatch often necessitates both hands, especially for continuous input such as flicking the device screen with the opposite-side hand (OSH) [34]. This becomes tedious as such wearable devices are predominantly valuable for glancing at information when the users' hands are occupied while holding objects or busy at other tasks.

Efforts are underway at developing methods to allow same-side hand (SSH) operation on smartwatches. However, these have primarily targeted discrete input operations, such as in the case of micro-interactions [21, 33] or for assigning commands to finger postures [10, 24, 36]. Tilting the wrist is a viable approach [9], but comes at the cost of quickly losing visual contact with the display as tilt movements can exceed the acceptable screen viewing ranges [9, 23]. Performing more expressive continuous gestural input still remains challenging using the same-side hand.

We study and present an alternative approach, WristWhirl, an interaction technique that uses continuous wrist movements, or whirls, for one-handed operation on smartwatches (Figure 1). When observing the collective range-of-motions of the wrist along each of its axes of movement [12] (see Figure 2 and the WRIST AS JOYSTICK section), the hand can be viewed as a natural joystick. We explore the ability of the human wrist to perform complex gestures using full wrist motions, or wrist whirls. We first demonstrate that wrist whirl is sufficiently expressive to capture common touch interactions as well as generate free-form shapes (Figure 1 right) without impacting screen viewing stability. To validate the use of WristWhirl in different application scenarios, we implemented a proof-of-concept wristband sensor (Figure 1 left) by augmenting the strap of a smartwatch using an array of infrared proximity sensors, facing the user's palm. The sensors detect the wrist's joystick-like motion by sensing the degree of flexion/extension and ulnar/radial deviation of the wrist motion. Our preliminary system evaluation showed that the user could use the prototype to draw gestures at a quality comparable to that achieved by a commercial motion tracking system (e.g. Vicon [3]). Our approach does not seek to replace two-handed use of smartwatches, but instead provides an alternative to same-sided smartwatch input.

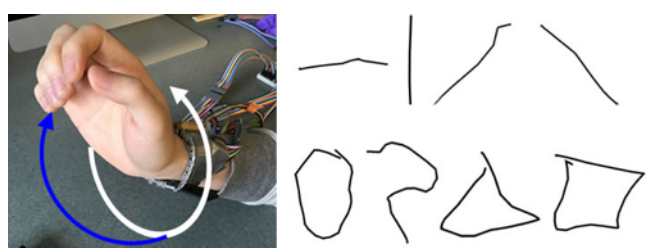

Figure 1. Left: Wrist whirling using our prototype. Right: example gestures drawn using the prototype [top: horizontal, vertical, slash, backslash, equivalent to flicking the touchscreen; bottom: circle, question mark, triangle, rectangle]

UIST '16, October 16-19, 2016, Tokyo, Japan
© 2016 ACM. ISBN 978-1-4503-4189-9/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984563

Our contributions from this work include: (1) the notion and investigation of using wrist whirls for one-handed continuous input on smartwatches; (2) the implementation and evaluation of a proof-of-concept prototype for detecting wrist whirl gestures; and (3) a set of usage scenarios to demonstrate WristWhirl's unique capabilities.

LITERATURE REVIEW

In this section, we present the existing literature in enabling one-handed interaction on smartwatches using discrete and continuous gestures. Given this scope, we exclude prior research on interactions without involving hand input (e.g. voice input). We also discuss various sensing techniques that have been developed in this context.

One-handed Discrete Gestures on Smartwatches

For the most part, research on one-handed input for smartwatches has focused on trigger discrete commands. Among this class includes techniques such as pinch (e.g. thumb touching the other fingers) [1, 4, 10, 13, 21, 26, 36] and different hand postures (e.g. fist or thumb-up) [10, 11, 24, 36]. A variety of sensing techniques have been developed to detect these gestures, many of which can be well integrated into a smartwatch form factor. Perhaps the earliest work in this category is GestureWrist [24], a technique that uses an array of capacitive sensors to detect the changes in forearm shape to inform different hand postures. Fukui, et al. [11] and Ortega-Avila et al. [22] demonstrated that forearm shape can also be sensed by using an array of infrared photo reflectors placed inside the wristband. More recently, WristFlex [10] and Tomo [36] showed that sensing capability can be improved by using force resisters or electrical impedance tomography (EIT) sensors.

Acoustic sensors have also been effective in detecting pinch gestures. For example, Skinput [13] uses an array of contact microphones (e.g. piezo sensors) worn on the upper arm to detect sound waves generated by the fingers tapping each other. Amento et al. [4] showed that a single piezo sensor placed in a wristband can help detect finger taps or rubs, similar to the gestures that can be detected by the commercial product, Aria [1]. Other approaches include using EMG sensors [2, 17, 26] and cameras [7, 21], which all require the sensor to be either worn on the upper arm or on other body parts, thus being less practical to smartwatch users.

An important aspect of using minimalist gestures, such as pinch or a simple hand posture to interact smartwatches is the ability to maintain a stable screen during the gesture to ensure constant visual contact with the display. Although, techniques like Android's wrist gesture or even shaking the watch may also be used to trigger commands, they are limited to eyes-free scenarios, where visual contact with the screen is non-essential. Any input technique for smartwatches needs to maintain screen stability for continuous feedback and interaction.

One-handed Continuous Gestures on Smartwatches

In contrast to the discrete gestural input, little work has produced techniques for one-handed continuous gestural input on smartwatches. While existing methods were originally developed for a different set of applications that may be used in this context, there are limitations which may prevent them from being used effectively by smartwatch users. For example, Crossan, et al. [9]'s work uses the smartwatch as a motion sensing device to track the degree of wrist pronation to control the movement of a cursor in 1D on a handheld device. To detect the same pronation gesture, Strohmeier, et al. [27] proposed to attach a pair of stretch sensors to the skin of the forearm, which can also be used to detect the bend motion of the wrist for 1D gestural input. Rahman et al. [23] systematically studied the number of distinguishable levels in each of the wrist tilt axes. In principle, the same concept can be applied to control a 2D cursor to allow the users to draw common touchscreen gestures by tilting the watch screen in the x and y axes. This is a technique that has been developed in the past for handheld devices [8, 14, 23, 30]. Note that tilting the body of the smartwatch may lead to loss of visual contact with the screen as it moves away from the user's view. This makes such an approach unusable for tasks requiring visual attention [23]. Similarly, translating the watch like a peephole display [16, 35] may also be used to control the cursor but the same problem remains unsolved. Additionally, moving the watch may largely impact task completion time [16]. We designed our technique to particularly overcome this limitation and allow the screen of the smartwatch to remain relatively stable when the gesture is being drawn.

Micro-gestures [21, 33] may be used for one-handed input as well. Resent research has shown the possibility of using a wrist-mounted camera [18, 29] to capture the movement of the thumb on the other fingers, which in principle can be used for drawing gestures. However, using a camera may significantly impact the form factor of the smartwatch and may drain the battery quickly. On the other hand, Soli [20] requires the sensor's active region to face the fingertips, thus not suitable for one-handed interaction in a smartwatch.

The most relevant work to our research is that of Voyles et al. [28], who proposed to use the wrist as a joystick for steering a robot. The authors developed a data glove equipped with magneto-resistive sensors to detect the joystick motion of the wrist movement. While wearing a data glove may be acceptable for domain specific applications (e.g. controlling the movement of a robot), it can be inconvenient and inappropriate for smartwatch users and general consumers. In contrast, the sensor we developed can be integrated into the watchband thus having much less impact on the smartwatch form-factor. Finally, there is a lack of understanding of the usability of the wrist's joystick motion for input on smartwatches and in different mobile environments (including mobility or distractions). Table 1 summarizes the existing work in the design space of one-handed input on smartwatch using wrist-worn sensors.

	Discrete Gesture	Continuous Gesture
Screen Unstable	Shake the watch Android Wrist Gesture	Pronation (1 DOF) [9, 27] Peephole (2DOF) [16, 35] Tilt the screen (2DOF) [8, 14, 23, 30]
Screen Stable	Finger pinch [1, 4, 10, 13, 21, 26, 36] Hand posture [10, 11, 22, 24, 36]	**WristWhirl (2DOF)**

Table 1 Existing work and design space of one-handed interaction on smartwatches.

WRIST AS JOYSTICK

The wrist is one of the most flexible joints in the human body. It can rotate along the forearm in both directions (e.g. pronation and supination). It can also bend along the plane of the palm (e.g. flexion and extension) or the one that is perpendicular (e.g. ulnar and radial deviations) to the palm (Figure 2). Previous studies suggests that the maximum range-of-motion for each moving axes are approximately 60° and 45° for flexion and extension respectively, 15° and 30° for ulnar and radial deviations respectively, and 65° and 60° for pronation and supination respectively [12]. These fairly wide ranges-of-motions could be used to turn the wrist into a "joystick" for smartwatches input using the same-side hand

The joystick motion available while whirling the wrist can be mapped to continuous events on a smartwatch, such as drawing uni-stroke gestures such as flicks or different shapes. Since the maximum range-of-motion of the wrist is highly asymmetric due to the constraints imposed by the structure of the tendons, muscles, and bones of the forearm, the ability to draw multiple gestures with varying levels of complexities needs careful examination.

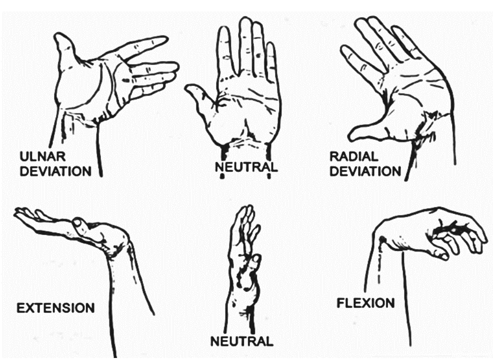

Figure 2. Whirling the wrist consists mainly of ulnar/radial deviations (top) and extension/flexion (bottom).

DESIGN CONSIDERATIONS

We present several factors that need to be considered for designing continuous one-handed input for smartwatches and which guided our exploration.

Screen Stability

Smartwatches already suffer from a limited display real-estate. Using the wrist as a joystick could mean rendering the screen quickly out-of-view for a given action. While it is impossible to completely eliminate screen movement when a gesture is drawn, our goal is to ensure this new input modality can minimize screen oscillations in comparison to methods that rely on tilting the watch for input. While other feedback modalities are possible, such as watch vibrations, when the gesture is accurately detected, we consider it important to have the screen in a reasonable viewing range to provide the same degree of fidelity as in touch interactions.

Eyes-free Input

Although screen viewing range is a key design consideration, many instances of smartwatch use could also benefit from eyes-free interaction. This could be in meetings, or during intense user activity, like a workout. Eyes-free input could also be driven by policies so as to minimize screen contact while driving or activities requiring the user's full attention. Therefore, intuitive mappings of wrist gestures to actions are needed to reduce user's dependence on full visual contact.

Control and Display Mapping

When an on-going gesture needs to be visualized on the watch display, the control and display mapping can be provided using either position- or rate-controlled mode. With position-control, the physical range of the wrist is mapped to actions on the screen, and the direction and amount of the wrist bend has a one-to-one mapping to the position of the on-screen visual cue (e.g. the trace of a gesture). Position control reinforces feedback via proprioception of the hand's orientation, allowing users to develop muscle memory for eyes-free interaction. In contrast, with rate-control, a cursor is needed and moves at a speed proportional to the direction and bend of the wrist. The cursor's rate of movement increases with the degree of wrist bend, i.e. more the hand is bent from its neutral pose, the faster the cursor moves. Rate-control mode is used in [28] but its control mechanism varies considerably from direct input on touchscreen devices. To minimize cognitive overhead in switching from direct touch to wrist gesture, we only explore position-control mode.

Gesture Delimiter

Explicit wrist gestures to trigger a smartwatch interaction need to be differentiated from normal wrist movements. Dwell can be used, but it can be less efficient. Another approach includes using only distinguishable movements that do not occur in normal day-to-day hand movements. While this method does not require a gesture delimiter, the number of usable wrist gestures is limited. Alternatively, a dedicated gesture can explicitly start and/or the end of a gesture. The delimiter can be a continuous gesture (e.g. a directional mark) or a discrete one (e.g. a finger pinch) but it needs to be reliable and easy to perform [25]. The latter approach requires an extra sensor for pinch detection but has the potential benefit of saving power (described later). In our implementation, we implemented the pinch delimiter using a simple sensing mechanism.

EXPLORING THE WRIST AS A JOYSTICK CONTROLLER

We conducted a study to investigate the bio-mechanical ability of the wrist to effectuate joystick-like gestures. We deem it an important first step to validate the feasibility of this new input method. We were particularly interested in measuring the efficiency and precision of such an input system as well as the amount of screen deviation caused by whirling the wrist.

Participants

Fifteen participants (2 females) between the ages of 20 and 30, all right-handed and daily computer users volunteered.

Gesture Set

To understand the ability to gesture using full wrist motion, we grouped gestures into two types [6]: 1) marking gestures; and, 2) free-form shape gestures. Marking gestures are directional strokes, are analogous to flicking a touchscreen, and are common for navigating large workspaces (e.g. a map or long list). Free-form shape gestures involve more complex shapes and can be rotationally invariant. For the directional marks, we included the horizontal and vertical strokes as well as two 45° strokes towards left and right (Figure 3 left). For the free-form path gestures, we chose four gestures from the gesture set shown to be useful on touchscreen devices [31, 32] (Figure 3 right). To ensure diversity, we picked the free-form gestures with straight lines and corners of different degrees (e.g. triangle and rectangle), one with a curvature path (e.g. circle) and one that is a mix of a curve, straight line, and corner (e.g. question mark).

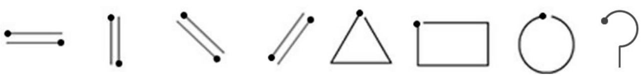

Figure 3. The eight tested unistroke gestures. The black dot indicates the start of the gesture.

Task and Procedure

In each trial, a gesture was shown to participants, who were then asked to reproduce the gesture as accurately and as fast as possible using their left wrist. A computer mouse was used on the right hand to indicate the start and end of a gesture just so that our study was not confounded by the implementation of the gesture delimiter (the delimiter we implemented in our prototype is described below, in the WRISTWHIRL PROTOTYPE section). Participants pressed and held the left mouse button to start drawing. Releasing the mouse button indicated the end of the gesture.

Participants were asked to perform the gestures in two different postures, hand-up and hand-down (Figure 4). In the hand-up condition, participants held the watch in front of their chest and with the hand-down condition, participants were required to have the watch hand hang naturally alongside the body. The former condition allows us to examine by how much the watch screen is titled during a whirl action, while the latter enables us to examine eyes-free input. When the watch was held in front of the chest, participants saw the gesture trace they were drawing on the watch screen. When the watch hand was hung alongside the body, no visual feedback was given, the gestures were drawn eyes-free. A computer monitor was placed in front of participants to show them the current gesture they needed to draw. The monitor turned blank after a trial started.

Finally, participants were also asked to perform the gestures while standing and walking. Similar to [6], in the Walking condition participants had to coordinate hand gestures while moving on a motorized treadmill at a speed of 3km per hour.

At the start of the experiment, participants were asked to practice gesturing using the wrist for as long as they wanted. Before each trial, one of the eight gestures was shown to the participant on both the watch and a monitor. On a left mouse button click, the watch display turned into an empty canvas with a black cursor on it. Participants were then instructed to hold down the mouse button and start drawing the gesture. Upon finishing the gesture (e.g. the mouse button was released), a new gesture was presented to the user. This process was repeated until all trials were completed at which point participants were asked to fill-in a questionnaire.

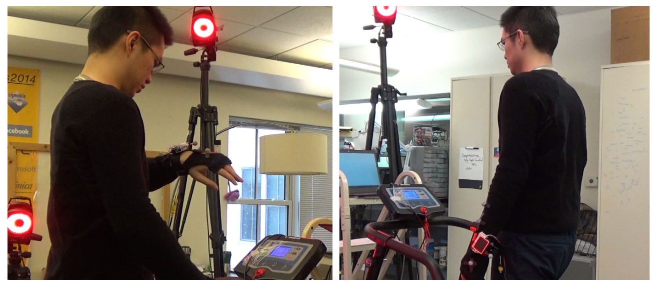

Figure 4. Hand postures: hand-up (left) and hand-down (right).

Apparatus

Wrist motion was captured using a Vicon motion tracking system (Figure 4) to ensure that results of the study are minimally affected by hardware implementation. The wrist gesture was transferred into the cursor movement on a 2D plane by projecting the position of the marker placed on the back of the hand onto the 2D plane perpendicular to the forearm. The trace of the cursor movement was shown on the watch screen as long at the mouse button was held. The mouse was mounted on the handle on the right side of the treadmill to make it easy for participants to reach. Finally, our custom-made smartwatch consisted of a 2" TFT display, used as an external monitor of a ThinkPad x1 Carbon laptop (Intel Core I7 2.1 GHz, 8 GB RAM) running the experiment software, which was written in C# .NET.

Study Design

The experiment employed a 2×2×2 within subject factorial design. In each trial, participants performed tasks in one of each *Gesture Type (mark or path)* × *Mobility (standing or walking)* × *Hand Posture (hand-up or hand-down)* combination. Each condition was repeated 10 times. For the conditions involving bidirectional marks, participants were asked to draw the mark in either direction (e.g. left to right or right to left in the horizontal condition) for half of the repetition. The conditions were counter-balanced among participants and the order of the gestures was randomized. The experiment design can be summarized as: *2 Gesture Types* × *4 gestures per type* × 2 *Mobility* × 2 *Hand Postures* × 10 R*epetitions* × 12 *Participants* = 3840 gestures.

Results and Discussion

We analyzed the data using a repeated-measures ANOVA and Bonferroni corrections for pair-wise comparisons.

Task Completion Time

Time was recorded when the mouse button was pressed and until the button was released. ANOVA yielded a significant effect of *Gesture Type* ($F_{1,11} = 276.37$, $p < 0.001$) and *Hand Posture* ($F_{1,11} = 6.9$, $p < 0.05$). There was no significant effect of *Mobility* ($F_{1,11} = 2.36$, $p = 0.153$). We found a significant interaction effect on *Gesture Type* × *Hand Posture* ($F_{1,11} = 9.43$, $p < 0.05$), indicating that a hand posture affected time differently for marks and free-form paths.

Overall, participants spent on average 960 ms per gesture. As expected directional marks required less time (483 ms, SE = 31) to draw than free-form paths (1436 ms, SE = 67) (Figure 5). An interesting finding is that task completion time was not affected by walking or standing but participants could perform wrist gestures faster with the hand alongside the leg (877 ms, SE = 41) than with the hand held in front of the chest (1043 ms, SE = 64). Our observation suggested that participants tended to slow down a bit to ensure that they could draw the gestures more precisely when they saw the visual feedback. This is particularly true for free-form paths.

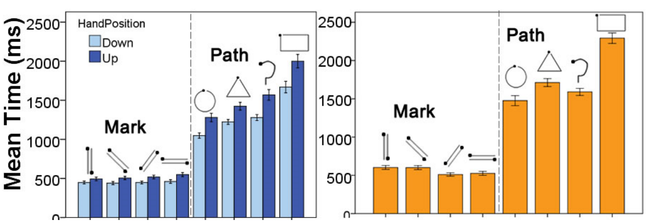

Figure 5. Task time shown by Gesture Type and Hand Posture. Task time completed using Vicon (left) and our prototype (right) (Error Bars show 95% CI in all figures).

Screen Deviation and Stability

Wrist gestures can lead to loss of visual contact with the smartwatch screen or cause a blurred view of the screen content. To assess the degree of screen movement and sway we captured two metrics: *screen translation* and *screen oscillation*. The a*verage translation distance T* (Equation 1) simply measures the amount of screen movement in 3D space during the course of a gesture. It is defined as the sum of the distances from the current screen position (p_i) and the screen position at the start of a gesture (p_0) over the course of the gesture, divided by the length of the gesture (n points).

$$T = \frac{\sum_{i=0}^{n} Dist(p_i,\ p_0)}{n} \qquad (1)$$

The a*verage screen oscillation O* (Equation 2) measures how much the path of screen changes direction over the course of a gesture. Considering the path of the screen as a series of between-point vectors, O is defined as the sum of the angle between the two adjacent vectors (v_i and v_{i+1}) over the course of the gesture, divided by the length of the path in n-1 vectors. If the screen keeps shaking (e.g. moving back and forth), it will continually change direction (e.g. 180°), which will lead to a very high O value.

$$O = \frac{\sum_{i=0}^{n} |Ang(V_i,\ v_{i+1})|}{n-1} \qquad (2)$$

Screen translation distance. ANOVA yielded a significant effect of *Gesture Type* ($F_{1,11} = 29.7$, $p < 0.001$) and *Mobility* ($F_{1,11} = 174.15$, $p < 0.001$). There was no significant effect of *Hand Position* ($F_{1,11} = 1.19$, $p = 0.3$). We found a significant interaction effect on *Gesture Type* × *Mobility* ($F_{1,11} = 31.86$, $p < 0.001$), which indicates that *Mobility* had more of a negative impact on translation distance for the free-form paths than directional marks. Overall, the average screen translation distance was 17 mm per gesture. The screen moved less with the directional marks (14 mm, SE = 1.5) than free-form paths (20 mm, SE = 1.3). It is worth noticing that most screen movements occurred while walking (24 mm, SE = 1.7) than standing (9 mm, SE = 1).

Since the major impact of screen translation occurs when the visual feedback is provided, we further investigate the effect of dependent variables when the hand was held in front of the chest. ANOVA revealed a similar trend as the one above. Overall, when the screen was held in front of the user, average translation distance was 16 mm. In particular, directional marks caused less translation (13 mm, SE = 1.5) than free-form paths (19 mm, SE = 1.2). There was also less translation in the standing condition (9 mm, SE = 1) than in the walking condition (23 mm, SE = 1.7) (Figure 6 left). Note that 23 mm deviation in the position of the watch still allows the screen to stay inside the user's view. However, if the screen shakes considerably, visual information becomes blurry. We thus looked into the stability of the screen using the screen oscillation metric.

Screen oscillation. We only analyzed the data when the hand was held in front of the chest. ANOVA yielded a marginal effect of *Gesture Type* ($F_{1,11} = 4.86$, $p = 0.05$) and significant effect of *Mobility* ($F_{1,11} = 8.18$, $p < 0.05$).

Overall, we found the screen oscillation to be 18°. This is the average change in the screen movement direction, which shows no back-and-forth movement of the watch screen caused by the wrist motion. In particular, the oscillation was slightly higher for the direction marks (18.7°, SE = 1.1) than for the free-form paths (17.4°, SE = 0.8). We also found less oscillation in the standing condition (14.7°, SE = 1.8) than in the walking condition (21.3°, SE = 1.1) (Figure 6 right).

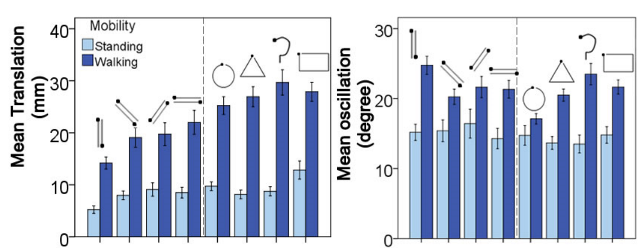

Figure 6. Mean translation distance and screen oscillation shown by Gesture Type and Mobility (Hand Position = Up)

Accuracy Analysis

We analyzed the gesture recognition accuracy for free-form paths and directional marks separately. For the free-form paths we were interested in the overall the shape of the gestures but for the directional marks we were also interested in the accuracy of their drawn direction as it is very common in smartwatch interaction (e.g. swipe).

Free-form path recognition accuracy. We used the $1 gesture recognizer [31] to measure the accuracy of the free-form paths drawn using WristWhirl. The result of a 12 fold cross-validation revealed that on average the $1 gesture recognizer was able to correctly recognize 93.8% of the gestures. Recognition accuracy was higher when the data was collected in the standing condition (95.1%) than in the walking condition (92.4%) ($p < 0.05$). There was no significant difference between the two hand postures ($p = 0.87$). Surprisingly, Question mark received the highest accuracy (100%), which was significantly higher than Circle (90.2%), Rectangle (90.8%), and Triangle (94%) (all $p < 0.001$). One reason might be that the Question mark gesture is not carried out at the limits of the wrist range-of-motion.

Effect of individual differences. To further investigate the consistency of the gestures drawn across all participants, we processed each participant's data through the gesture recognizer trained with the remaining participants' data. The result showed an average accuracy of 92.2%. This value is similar to that obtained above (only slightly lower than the overall accuracy of 93.8%), suggesting that the tested gestures could be drawn correctly using the wrist as a joystick. To further confirm how well the gestures are drawn, we manually inspect their visual appearances.

Gesture visual quality. The gesture recognizer can only distinguish different gestures without knowing if they are drawn consistently right or wrong. Therefore, we recruited three paid volunteers to visually inspect the 1920 free-form paths that were collected from the study. The inspectors were unaware of the purpose of the study and were asked to identify the shape of the gestures in isolation of each other. For each gesture, the inspectors had to choose one that best matched the shape of the presented gesture from a list of eight figures, among which four of them were distractors with similar shapes. For example, "7" was chosen to confuse with the "Question mark", "diamond" was chosen to confuse with the "Rectangle", "b" and "Pigtail" were chosen to confuse with the "Triangle" and "Circle". Adding these distractors allowed us to further ensure the quality of the correctly identified gestures.

The result showed that the inspectors were able to correctly identify 85% of the gestures. Notice that 74% of the errors occurred when the distractors were chosen instead of the desired gesture. For example, the "Pigtail" was chosen instead of the "Circle" when the two ends of the path crossed each other. Figure 7 shows an example of the common missed interpretation of the free-form gestures.

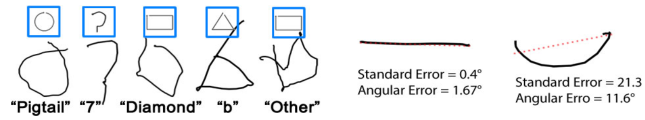

Figure 7. Examples of miss interpreted gestures by the inspectors (left). S_e in relation to two horiztontal marks (right). The red dotted line is the fitted model.

Accuracy of directional marks. We evaluated the accuracy of directional marks based on how straight the marks were drawn as well as how close the marks were to the desired direction (e.g. horizontal, vertical, 45° slash and backslash). For each directional mark, we used a linear regression to generate a straight line to fit the points of the gesture. We then used the standard error (S_e) of the regression (Equation 3) to describe how well the model has fitted the data. The smaller the value of S_e, the better the fit. For example, S_e equals zero when the mark is a straight line. Finally, we calculated the absolute value of the angle between the generated model and the ideal mark to measure angular error.

$$S_e = \sqrt{\frac{1}{N-2}\sum_{i=1}^{N}(y_i - \hat{y}_i^2)^2} \tag{3}$$

Overall, the result revealed an average standard regression error of 5.4 and an average angular error of ±9.4°. Pair-wise comparisons showed that participants could draw the vertical mark more straight ($S_e = 2.9$) than the horizontal mark ($S_e = 4.6$), both of which were more straight than the slash ($S_e = 6.8$) and backslash mark ($S_e = 7.35$) (all $p < 0.05$ except for the slash and backslash). While hand position did not have a significant effect on S_e ($p = 0.53$), participants were able to draw directional marks more straight when standing ($S_e = 4.9$) than walking ($S_e = 5.9$) ($p < 0.05$). Figure 7 (right) gives a brief idea about the relationship between S_e and the "straightness" of two horizontal marks.

With respect to the angular error, vertical marks had the least angular error (±6.14°) ($p < 0.05$), followed by the horizontal (±9.6°), backslash (±9.8°) and slash (±11.9°) marks. No significant difference was found among them (all $p > 0.05$). We also found no significant difference between the two mobility conditions ($p = 0.14$) and hand postures ($p = 0.12$). Overall, this result highlights the ability of participants to draw straight lines with good accuracy in most direction by using the wrist as a joystick.

Subjective Ratings

To assess the physical exertion of using WristWhirl to perform gestures, participants were asked to rate each gesture on the Borg CR10 Scale [5]. Overall, the directional marks were rated easy to perform (avg. = 2.1) whereas the free-form paths were rated moderately (avg. = 3.6) (Figure 8). Swiping vertically was considered very easy as all the participants rated it lower than 2. Rectangle was considered somewhat difficult where more than 58% of our participants gave it a higher grade than 5 (e.g. hard). Figure 8 shows the ratings for all the eight gestures. We also asked participants to rate the acceptance of WristWhirl in different settings. The result showed that participants considered it socially acceptable to use wrist gestures in front of people (3, with 1 being strongly acceptable and 10 being strongly unacceptable) although they felt more comfortable to use wrist gestures in private (avg.=1.4).

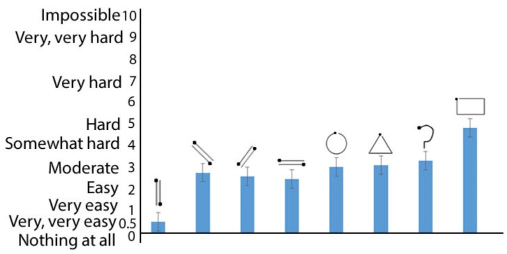

Figure 8. Perceived exertion rating of the tested wrist gestures.

Overall, the promising results show that common touchscreen gestures can be drawn using the wrist's joystick motion for occasional use. We also demonstrated that drawing gestures did not lead to significant screen deviation in its position. Nor did we find that the screen shook significantly during the course of a gesture. An interesting observation is that muscle memory was commonly used by participants in guiding their wrist trajectories. Participants commented that while visual feedback was definitely helpful in the training phase to help them develop correct muscle memory, it becomes less important when they know how to draw with wrist. This might explain why hand posture did not affect the quality of the gestures as much as we expected.

DEVICE IMPLEMENTATION OPTIONS

Results from the first study led to exploring sensor alternatives for realizing the range of potential wrist whirl motions. We aimed at developing a self-contained, smartwatch form-factor prototype. In this section, we present the device's implementation options in terms of different sensor options, sensing resolution, and options for sensing delimiter. We also discuss the advantages and disadvantages of the various hardware alternatives.

Sensor Options

Several options exist for choosing an appropriate sensing mechanism to enable our input mechanism. For example, an array of proximity sensors (either infrared or ultrasonic proximity sensor) placed on the watch strap can be used to detect the flexion, deviation, and extension of the wrist by measuring how close the bent hand is to the watch. Alternatively, strain gauges can also detect how much the wrist is bent. However, strain gauges require physical contact with the base of the hand to sense the bend motion, thus requires the watch to be placed close to the palm. This could lead to discomfort. Additionally, sensing accuracy depends on the proximity of the strain gauge to the base of the hand. As a watch's position always shift during use, this approach is prone to errors. Cameras might also be used to detect wrist motion. Similar to proximity sensors, the placement of cameras along the forearm is not constrained to how close the sensor is placed to the base of the hand. However, running multiple cameras and processing video streams may consume significantly more power than the above options. After considering the pros and cons of the above options, we decided to use the proximity sensor.

Sensing Resolution

Fine-grained sensing resolution is preferred but it would be difficult to achieve the level of resolution of a Vicon motion tracking system. With the existing sensor options, each single proximity or strain gauge sensor can serve as a sensing pixel. Thus, the sensing resolution of the final unit is also dependent on the physical size of the sensor: the smaller the sensor, the more can be installed on the strap, to provide a higher resolution. In reality a compromise is necessary to achieve a balance between the sensing resolution, the smartwatch physical form factor, and power consumption.

Gesture Delimiter

We decided to use a dedicated delimiter sensor to detect the start and end of a gesture (e.g. finger pinch). Using a decided delimiter sensor can lead to significant power conservation as it allows the wrist motion sensors to be only turned on when a pinch is detected. The motion sensors can be turned off upon the end of a gesture. The pinch sensor needs to be self-contained in the smartwatch form factor thus requiring a small size. Few options exist for such a sensor. For example, the smartwatch's built-in IMU sensor may be used to detect the pinch gesture but it may be power consuming and prone to motion noise. A skin-contact piezo is equally an option. The piezo sensor detects the sound of the finger pinch propagating through the user's skin [4], which has been shown effective in detecting pinch in an arm band form factor [13]. Piezo is also extremely efficient on battery life. We thus decided to use a piezo.

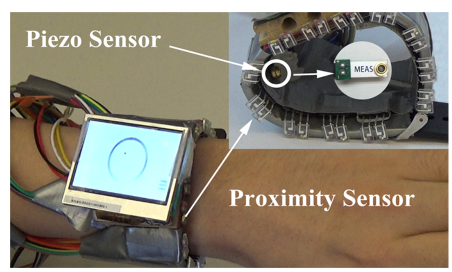

Figure 9. The WristWhirl prototype.

WRISTWHIRL PROTOTYPE

To explore one-handed interactions enabled by wrist whirl gestures we created a proof-of-concept system, WristWhirl. The prototype is made of a 2" TFT display and a plastic watch strap augmented with 12 infrared proximity sensors, each composed of a pair of IR emitters and detectors (LITON LTE-301 & 302), placed on the strap in approximately 0.4 cms apart from each other (Figure 9). The proximity sensor operates at 940nm, thus differentiating its signal from visible light. It has a maximum sensing distance of approximately 12 cm. Our test showed that adjacent sensors did not interfere with one another. The sensors were connected to an Arduino DUE board, which was then connected to a Lenovo ThinkPad x1 Carbon laptop, reading the sensor data at a speed of 9600 Hz. The Arduino provides readings from 0 to 1023 with 1023 being the closest proximity.

Pinch detection was implemented using a piezo vibration sensor (Minisense 100) placed inside the wrist strap (Figure 9). The user can pinch to indicate the start of a gesture, which turns on the proximity sensors to capture the wrist motion. Upon finishing the gesture, the user can do another pinch to indicate the end of the gesture. This turns off the proximity sensors to save battery.

Device calibration. Calibration is needed for different lighting conditions. The user needs to rotate the wrist in a circular motion similar to drawing a circle, at least once (more rotations only incrementally improves recognition). This process calibrates the sensor with the maximum range of motion for each of the wrist's moving axes.

Since the wrist's maximum range of motion is asymmetric along different axes, performing the circular motion by banding the wrist to its limit will result in a kidney-shaped gesture rather than a circle. It is thus a design decision whether we want to keep the resulting gesture a kidney shape or map it to a circle, in which case, the points drawn near the boundary of the wrist's range of motion will be scaled towards that circle, making the gesture look slightly stretched. We decided to go with the circle as it may be what people expect to see.

Another purpose of the calibration is to normalize the sensor readings with the magnitude of the inferred noise in the environment. We implemented a simple method to allow the user to skip the calibration phase if the environmental noise is similar to a previously recorded value (e.g. ±20 of each sensor's reading). This way the system can use the data from a previous calibration. Therefore, recalibration is only needed when lighting conditions change significantly.

Tracking algorithm. We treat the data from each proximity sensor as a vector, the direction of which is determined by the location of the sensor along the watch band. The length of the vector is determined by the value of the sensor. The higher the value, the longer the vector. The direction and how much the wrist is bent is detected from the sensor with the highest reading. As the sensors were placed in close proximity to each other, it is almost the case that more than one sensor can observe very high readings. In this case, we take the data from three consecutive sensors, which in total provides the highest value among all the consecutive triplets. We then take the summary of the three corresponding vectors, the direction of which estimates the tilt direction of the wrist. The length of this vector will exceed the highest reading of the proximity sensor (e.g. 1023). We adjust its length based on the readings of its two direct adjacent sensors using linear interpolation. The end point of the resulting vector is the position of the wrist in 2D space (represented by a cursor). The hand's neutral position is detected when the greatest difference among the sensor values becomes lower than a threshold (e.g. 50), in which case, the resulting vector is the sum of all 12 sensor vectors and the cursor stays near the middle of its active region. We found this simple method worked well to estimate the wrist's joystick motion while the user may need to adjust the wrist movement to accommodate this slightly different control-to-display mode.

WRISTWHIRL USAGE SCENARIOS

We implemented four applications using off-the-shelf games and Google Maps to illustrate the potential usage scenarios of WristWhirl. All the applications require continuous input and are normally used by both hands. Gesture Shortcut showcases gesture recognition, and the other applications all take the advantage of two dimensional continuous control to provide a richer and more expressive interaction.

Gesture Shortcuts

We implemented a gesture shortcut app (Figure 10), which allows the user to launch favorite smartwatch applications by drawing gestures. This is similar to the popular gesture search app on smartphones [19]. However, we allow it to be used on the smartwatch by using one-handed interaction. In our current implementation, the user can launch the calendar app by "drawing" a triangle. Similarly, the user can use gestures to speed dial a number. For example, the user can draw an "L" to call Lisa.

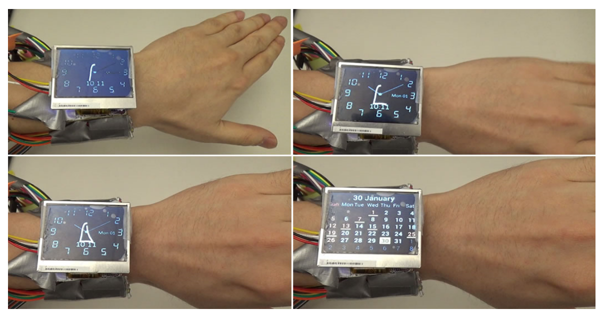

Figure 10. Drawing a triangle to launch a calendar app.

Music Player

Using discrete commands [1] to navigate a long list of songs can be tedious as only one item in the list can be advanced per action. With WristWhirl a long wrist-swipe allows the user to quickly skip a number of songs whereas a short wrist-swipe advances one song at a time. We implemented a music player app, in which the users can use wrist extension/flexion to scroll a list of songs (Figure 11). The user can double tap the thumb and index finger to play the selected song. Notice that we use double tap to distinguish between selection and a gesture delimiter (a single pinch). This approach to scrolling can allow the user to navigate a list eyes-free, with the simple addition of audio feedback.

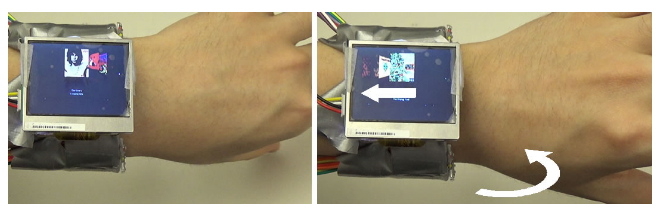

Figure 11. Wrist extension flips a list of songs to the left

2D Navigation

WristWhirl allows 2D panning and zooming by using one hand. In our implementation of a map application, the user can use wrist ulnar/radial deviation to pan up or down and extension/flexion to pan left or right. That is when the watch screen is held horizontally in front of the chest, gesturing towards the body pans the map down, gesturing upwards pans the map left, and vice versa (Figure 12). The user can control the panning distance with the length of a gesture. Whirling the wrist in the counter-clockwise direction zooms in the map. Alternatively, whirling the wrist in the clockwise direction zooms out. Double tapping the thumb and index switches between the two modes. The user can use the same interaction technique to navigate a photo album or webpage.

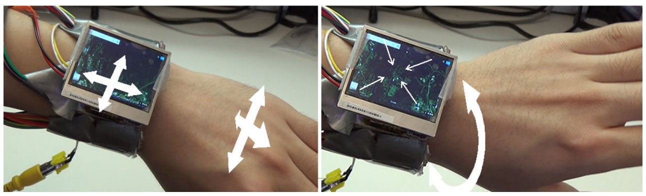

Figure 12. Panning achieved with ulnar/radial deviation (left) and zooming a map made possible by clockwise and counter-clockwise whirls (right).

Game Input

Playing games often requires continuous input for the best gaming experience. In our implementation, the user can play Tetris by swiping the wrist left and right. Wrist extension is used to change orientation and wrist flexion is used to drop the piece (Figure 13). Notice that the user's dominant hand is now free to perform simple tasks, such as picking up a phone, without interrupting the game. This type of input can also be used for other games. For example, the user can whirl the wrist to play Fruit Ninja.

Figure 13. Playing Fruit Ninja (left) and Tetris (right)

PRELIMINARY SYSTEM EVALUATION

We conducted a preliminary system evaluation to verify the accuracy of our prototype. We were interested in knowing how well users employ our prototype to create gestures. Ideally the evaluation would be conducted using the VICON motion tracking system as a baseline. We found it difficult in practice as both system use infrared for illumination. As a result, the infrared light from the VICON interfere with the proximity sensors of our prototype. At the current stage, we decided to only evaluate our prototype by measuring how well the gestures can be drawn by using our prototype.

Participants and Apparatus

We recruited 12 participants between the age of 20 and 30 (10 male). 7 of them participated in Study 1. We used the same setup as in Study 1, but instead of the Vicon our prototype tracked the wrist motion.

Task and Procedure

Participants were asked to perform the same set of gestures as in Study 1 except that they only did the study in the standing position with their hand being held in front of the chest. At the beginning of the study, participants had to calibrate the prototype and practice for about 20 minutes. The *Gesture Type* was counter-balanced among participants. The study had the same procedure as in Study 1 except that at the end of the study, participants had the opportunity to try five demo apps discussed above and provide feedback.

Result

Task completion time. On average, it took the participants 560 ms to perform the directional marks and 1767 ms to perform the free-form gestures. A comparison of the task completion time between the WristWhirl and Vicon (both in hand-up and standing conditions) using an independent-sampled t-test revealed a marginal difference for the directional marks (e.g. Vicon: 546 ms) ($t_{958} = 1.29$, $p = 0.05$) and significant difference in the free-form path gestures (e.g. Vicon: 1564 ms) ($t_{958} = 6.67$, $p < 0.001$). WristWhirl was slower than Vicon because our prototype was not as sensitive as the Vicon when the wrist was near the natural position (e.g. slightly tilted). Participants thus needed to slightly exert more tilt in our prototype for the sensor to pick up the wrist motion. This led to larger gestures and longer task completion times. Figure 5 shows a side-to-side comparison of the gesture completion time using the VICON motion tracking system (left) and our prototype device (right).

Free-form path recognition accuracy. The result of a 12 fold cross-validation showed that on average the $1 gesture recognizer was able to correctly recognize 95.4% of the free-form paths drawn using our prototype. A t-test showed that the recognition accuracy for WirstWhirl is significantly higher than for the Vicon (92.5%; $t_{958} = 1.89$, $p < 0.001$).

Accuracy of directional marks. With respect to the accuracy of the directional marks (e.g. straightness), the result showed an average standard error of regression of 1.56, which is significantly lower than the Vicon (5.23, $t_{958} = -17.58$, $p < 0.001$). The average angular error was 7.3°, which is also significantly lower than the Vicon (9.8°, $t_{958} = -5.18$, $p < 0.001$). Figure 1 right shows an example of a few gestures collected in this evaluation.

These results suggest a comparable performance of our prototype as with the Vicon. We attribute this advantage to a mixed reasons. Learning from the first study could be one factor but it should be minimal as the two studies took place three weeks apart. The prototype also used a different tracking algorithm, which may also contribute to accuracy.

Subjective feedback. Overall, participants welcome the idea of using the wrist gesture for one-handed interaction on smartwatches. While some of them felt it a bit awkward to use at the beginning, they all liked it after they learned how the device operated. A participant commented that *"I think it is quite easy to use" (P8).* As expected participants thought it would not be very comfortable to exert the wrist for long time periods but all see the value of WristWhirl as an alternative input method for occasional use. A participant said "*It is very helpful when the other hand is carrying some very heavy bags!*" *(P5).* Participants also enjoyed our demo apps and saw themselves using some of the apps in their daily life. For example, a participant said that *"I like the map application very much" (P7).* Another one said that the *"Music Player is so cool and helpful!" (P1)*. Most of the participants preferred the simple directional gestures over the free-form paths. They all liked the Fruit Ninja app but most preferred playing the game on a larger touchscreen device. They could also envision playing Tetris with WristWhirl.

DISCUSSION AND LIMITATIONS

We discuss the insights gained from this work, the lessons we learned, the limitations of our approach, and present directions for future research.

Learnability

While many of our daily activities already involve wrist motion with various degrees of complexity or cognitive levels (e.g. low when using a spatula and high when controlling the swing of a tennis racquet), drawing touchscreen gestures using the wrist's joystick motion is not something people can master without learning. This is mainly attributed to the inconsistency between people's perceived gesture that a certain wrist motion may produce and the actual gesture the wrist motion produces. For example, a horizontal line often ended up being drawn as a flat "v" shape in the initial stages of training as participants did not realize that they were moving the wrist in a curved trajectory. This is also due to the lack of visual reference on the forearm to guide the movement of the wrist in a desired way. The outcome of learning, however, is noticeably encouraging. Participants were excited about how well they can draw the touchscreen gestures with the wrist. For example, a participant commented that *"I am amazed by how much I can do with my wrist" (P3)* and another participant said that *"I now see myself using it to interact with a watch" (P11)*. To reduce training length one could adapt the system to match what users 'think' they are drawing (e.g. a horizontal line can be produced if the user is drawing a flat "v"). Future work will explore this direction.

User Evaluation

The presented user evaluation is limited in that we only tested a small set of common free-form paths. Future research will study more different paths (e.g. in curved/spiral shapes). While the goal of our study was to show evidence to support wrist whirls as a new input modality, more work is needed to understand the usability of this input style in real-world practice, in which unexpected uncertainties may influence the result and may possibly lead to a different conclusion. Finally, a longer-term study can help tease the memorability of wrist gestures.

System Implementation and evaluation

The proximity sensor is robust against visible light but it could be interfered with by the infra-red noise in the environment. In addition to natural daylight, there exists many infra-red light sources in office and home environments (e.g. security cameras). A possible way to avoid the inference from ambient light is to modulate the light signal in a certain frequency. Future research will test this method in a real-world environment. Additionally, our current implementation only works for one particular wrist size. We will explore alternative design options to facilitate a wide range of input with different wrist sizes. Pinch detection can also be improved. The current method may trigger false pinch events when the index taps a hard surface such as typing on a keyboard. Future work will focus on studying different delimiter options. Finally extra haptic feedback, such as Skin Drag Displays [15], can further facilitate eyes-free input.

Multi-touch gestures

Our system does not support multi-touch gestures. Therefore, common gestures such as two-finger scrolls cannot be performed using WristWhirl. Future research will explore potential methods that can enable multi-touch style continuous input using one hand.

CONCLUSION

One-handed interaction on smartwatches is challenging as existing ways of using discrete input actions, such as pinch, do not support 2D continuous gestural input. While other approaches such as tilting the watch may be used for continuous input, such approaches are prone to losing visual contact with the display when the screen is tilted away from the user. In this paper, we propose to use the wrist as an always-available joystick to perform common touchscreen gestures using the same-side hand wearing the watch. We describe a number of design considerations in designing this new input style. Through a user study we measure how fast and precise gestures can be drawn using the wrist's joystick motion in two hand postures and while walking or standing still. We also measured the amount of screen movement during the course of a gesture. The results we obtained from observing the bio-mechanical influences of wrist whirls led to the development of a proof-of-concept prototype in the form of a wristband, with which we demonstrated a number of applications that can potentially benefit from one-handed continuous input on smartwatches. We believe that our work serves as important groundwork for exploring one-handed interaction techniques on smartwatches.

REFERENCES

1. Aria Wearable, https://www.ariawearable.com/
2. Myo Gesture Control Armband, https://www.myo.com/
3. VICON Motion Tracking System, http://www.vicon.com/
4. Amento, B., Hill, W., and Terveen, L. 2002. The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02 Extended Abstracts on*

Human Factors in Computing Systems (CHI EA '02). ACM, New York, NY, USA, 724-725. DOI=http://dx.doi.org/10.1145/506443.506566

5. Borg, G. 1970. Perceived exertion as an indicator of somatic stress. *Scand j Rehabil Med 2(3),* 92-98.
6. Bragdon, A., Nelson, E., Li, Y., and Hinckley, K. 2011. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 403-412. DOI=http://dx.doi.org/10.1145/1978942.1979000
7. Chan, L., Chen, Y., Hsieh, C., Liang, R., and Chen, B. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). ACM, New York, NY, USA, 549-556. DOI=http://dx.doi.org/10.1145/2807442.2807450
8. Crossan, A., and Murray-Smith, R. 2004. Variability in Wrist-Tilt Accelerometer Based Gesture Interfaces. In *Proceedings of the 6th international conference on Human computer interaction with mobile devices and services* (MobileHCI '04). ACM, Glasgow, Scotland, 144-155.
9. Crossan, A., Williamson, J., Brewster, S., and Murray-Smith, R. 2008. Wrist rotation for interaction in mobile contexts. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services* (MobileHCI '08). ACM, New York, NY, USA, 435-438. DOI=http://dx.doi.org/10.1145/1409240.1409307
10. Dementyev, A., and Paradiso, J.A. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). ACM, New York, NY, USA, 161-166. DOI=10.1145/2642918.2647396 http://doi.acm.org/10.1145/2642918.2647396
11. Fukui, R., Watanabe, M., Gyota, T., Shimosaka, M., and Sato, T. 2011. Hand shape classification with a wrist contour sensor: development of a prototype device. In *Proceedings of the 13th international conference on Ubiquitous computing* (UbiComp '11). ACM, New York, NY, USA, 311-314. DOI=http://dx.doi.org/10.1145/2030112.2030154
12. Grandjean, E. 1969. Fitting the task to the man: an ergonomic approach Taylor and Francis. 372.
13. Harrison, C., Tan, D., and Morris, D. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10). ACM, New York, NY, USA, 453-462. DOI=http://dx.doi.org/10.1145/1753326.1753394
14. Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. 2000. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology* (UIST '00). ACM, New York, NY, USA, 91-100. DOI=http://dx.doi.org/10.1145/354401.354417
15. Ion, A., Wang, E.J., and Baudisch, P. 2015. Skin Drag Displays: Dragging a Physical Tactor across the User's Skin Produces a Stronger Tactile Stimulus than Vibrotactile. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 2501-2504. DOI=http://dx.doi.org/10.1145/2702123.2702459
16. Kerber, F., Krüger, A., and Löchtefeld, M. 2014. Investigating the effectiveness of peephole interaction for smartwatches in a map navigation task. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services* (MobileHCI '14). ACM, New York, NY, USA, 291-294. DOI=http://dx.doi.org/10.1145/2628363.2628393
17. Kerber, F., Lessel, P., and Krüger, A. 2015. Same-side Hand Interactions with Arm-placed Devices Using EMG. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '15). ACM, New York, NY, USA, 1367-1372. DOI=http://dx.doi.org/10.1145/2702613.2732895
18. Kim, D., Hilliges, O., Izadi, S., Butler, A.D., Chen, J., Oikonomidis, I., and Olivier, P. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, 167-176. DOI=http://dx.doi.org/10.1145/2380116.2380139
19. Li, Y. 2010. Gesture search: a tool for fast mobile data access. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (UIST '10). ACM, New York, NY, USA, 87-96. DOI=http://dx.doi.org/10.1145/1866029.1866044
20. Lien, J., Gillian, N., Karagozler, M.E., Amihood, P., Schwesig, C., Olson, E., Raja, H., and Poupyrev, I. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. In *Proceedings of the 43rd ACM Special Interest Group on Computer Graphics and Interactive Techniques* (SIGGRAPH'16). ACM, Anaheim, CA, USA.

    DOI=http://dx.doi.org/10.1145/2897824.2925953
21. Loclair, C., Gustafson, S., and Baudisch, P. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proc. MobileHCI Workshop on*

Ensembles of On-Body Devices. Lisbon, Portugal, 4 pages.

22. Ortega-Avila, S., Rakova, B., Sadi, S., and Mistry, P. 2015. Non-invasive optical detection of hand gestures. In *Proceedings of the 6th Augmented Human International Conference* (AH '15). ACM, New York, NY, USA, 179-180. DOI=http://dx.doi.org/10.1145/2735711.2735801
23. Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. 2009. Tilt techniques: investigating the dexterity of wrist-based input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 1943-1952. DOI=http://dx.doi.org/10.1145/1518701.1518997
24. Rekimoto, J. 2001. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers* (ISWC '01). IEEE Computer Society, Washington, DC, USA, 21-.
25. Ruiz, J., and Li, Y. 2011. DoubleFlip: a motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 2717-2720. DOI=http://dx.doi.org/10.1145/1978942.1979341
26. Saponas, T.S., Tan, D.S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J.A. 2009. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09). ACM, New York, NY, USA, 167-176. DOI=http://dx.doi.org/10.1145/1622176.1622208
27. Strohmeier, P., Vertegaal, R., and Girouard, A. 2012. With a flick of the wrist: stretch sensors as lightweight input for mobile devices. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (TEI '12), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 307-308. DOI=http://dx.doi.org/10.1145/2148131.2148195
28. Voyles, R., Bae, J., and Godzdanker, R. 2008. The gestural joystick and the efficacy of the path tortuosity metric for human/robot interaction. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (PerMIS '08). ACM, New York, NY, USA, 91-97. DOI=http://dx.doi.org/10.1145/1774674.1774689
29. Way, D. and Paradiso, J. 2014. A Usability User Study Concerning Free-Hand Microgesture and Wrist-Worn Sensors. In *Proceedings of the 2014 11th International Conference on Wearable and Implantable Body Sensor Networks* (BSN '14). IEEE Computer Society, Washington, DC, USA, 138-142. DOI=http://dx.doi.org/10.1109/BSN.2014.32
30. Weberg, L., Brange, T. and Hansson, Å.W. 2001. A piece of butter on the PDA display. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '01). ACM, New York, NY, USA, 435-436. DOI=http://dx.doi.org/10.1145/634067.634320
31. Wobbrock, J.O., Wilson, A.D., and Li, Y. 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
32. Wobbrock, J.O., Morris, M.R., and Wilson, A.D. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 1083-1092. DOI=http://dx.doi.org/10.1145/1518701.1518866
33. Wolf, K., Naumann, A., Rohs, M., and Muller, J. 2011. Taxonomy of microinteractions: defining microgestures based on ergonomic and scenario-dependent requirements. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction* (INTERACT '11). Lisbon, Portugal, 559-575.
34. Xiao, R., Laput, G., and Harrison, C. 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
35. Yee, K. 2003. Peephole displays: pen interaction on spatially aware handheld computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03). ACM, New York, NY, USA, 1-8. DOI=http://dx.doi.org/10.1145/642611.642613
36. Zhang, Y., and Harrison, C. 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). ACM, New York, NY, USA, 167-173. DOI=http://dx.doi.org/10.1145/2807442.2807480