Float: One-Handed and Touch-Free Target Selection on Smartwatches

Ke Sun, Yuntao Wang†, Chun Yu, Yukang Yan, Hongyi Wen, Yuanchun Shi
Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
{k-sun14, yyk15}@mails.tsinghua.edu.cn, {yuntaowang, chunyu, shiyc}@tsinghua.edu.cn

ABSTRACT
Touch interaction on smartwatches suffers from the awkwardness of having to use two hands and the "fat finger" problem. We present Float, a *wrist-to-finger* input approach that enables one-handed and touch-free target selection on smartwatches with high efficiency and precision using only commercially-available built-in sensors. With Float, a user tilts the wrist to *point* and performs an in-air finger tap to *click*. To realize Float, we first explore the appropriate motion space for wrist tilt and determine the *clicking* action (finger tap) through a user-elicitation study. We combine the photoplethysmogram (PPG) signal with accelerometer and gyroscope to detect finger taps with a recall of 97.9% and a false discovery rate of 0.4%. Experiments show that using just one hand, Float allows users to acquire targets with size ranging from 2mm to 10mm in less than 2s to 1s, meanwhile achieve much higher accuracy than direct touch in both stationary (>98.9%) and walking (>71.5%) contexts.

Author Keywords
Smartwatch; one-handed interaction; tilt; finger gesture; target selection

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User interfaces; Input devices and strategies

INTRODUCTION
Smartwatches are becoming more popular with users, providing efficient interactions on quick information access and response. Touch is still the most dominant way users interact with smartwatches, which benefits from its nature of direct input and the transfer of usage from other modern touchscreen devices. After a user survey and literature investigation, we however observe two typical problems of using touch to interact on smartwatches affecting its user experience.

† denotes the corresponding author.

CHI 2017, May 06-11, 2017, Denver, CO, USA
© 2017 ACM. ISBN 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3026027

Figure 1. Potential scenarios of Float. Left: a user handles an incoming call while riding by tilting the wrist and clicking the "message" button with a finger tap; Right: a user is checking the shopping list on a smartwatch while carrying lots of bags.

First, touch on smartwatches always requires the "free hand" or the "other hand" on which the smartwatch is not attached. Interactions such as carrying objects, cooking, holding an umbrella, riding a bike and driving will be inconvenient when the other hand is occupied; The second drawback is that of the fat finger problem, which is exacerbated by the ultra-small display on smartwatches. The finger can easily occlude over half of the display [45]. Commercial products enlarge selectable items to ensure the accuracy, making many of the basic interactions slow and laborious with sequences of swipes and taps [21].

Based on these two limitations, we are motivated to seek out alternative input modalities that enable one-handed and non-touch interactions on smartwatches. Our goal was to provide more hands-free interactions, while improving the accuracy and efficiency of target selection beyond direct touch. In this paper, we propose Float, an interaction technique that enables one-handed and touch-free input on smartwatches based on a combination of wrist tilt and finger gestures. The basic idea is simple: a user tilts the wrist to *point* and performs an in-air finger tap to *click*. For example, a user can handle incoming calls with a single hand while riding a bicycle (Figure 1.a).

Tilt is not novel. Prior research has investigated tilt for menu navigation [40, 26], game control and text entry [32, 42] on mobile phones and tablets. Most of the work however used tilt to perform only uniaxial selection by either mapping the tilt inclination or the tilt direction. With a smartwatch on the wrist, Float explores wrist tilt as a general pointing method over the watch face. This brings two benefits: (1) One can access any location on the watch face, which can be applied directly to existing devices with-out redesigning GUI into ad-hoc forms; (2) target selection can be achieved with only one hand, which is useful when the other hand is occupied. We also present a rigorous calculation to map users' wrist tilt movement to positions on the watch face based on the intrinsic rotation matrix.

After establishing tilt as the way of *pointing*, we first conducted a pilot experiment. In the first part, we explored the appropriate rotation space when tilting a wrist-worn watch; in the second part, we performed a user-elicitation study to determine the *clicking* interaction. Finally, finger motion/gesture of the involved hand was selected from five candidates, including dwell-time, voice, gaze, etc.

To realize the interactive system, we use only commercially built-in sensors on smartwatches. When a user performs a finger tap, the joint tendons and skin will cause significant motion and strain, which can be reflected by a peak in the accelerometer and gyroscope sequence of the motion sensor after a high-pass filter and a sharp *pulse-like* pattern in the PPG signal of the heart rate monitor. We implemented a detection algorithm combining thresholding, dynamic time warping and a k-NN classifier. To our knowledge, there is no previous work that detects hand actions using the PPG signal. Evaluation showed that finger taps could be detected with a recall of 97.9% and a false discovery rate of 0.4% in standing and walking contexts.

We conducted two user studies to evaluate the performance of Float. In the first study, we investigated the effects of target location (direction and distance) on pointing speed and stability through a discrete pointing experiment. We provide implications for layout and task design. In the second study, we conducted a continuous target selection experiment in both standing and walking contexts. We found that participants could select a target within 1s ~ 2s for target size ranging from 10mm to 2mm. In addition, the occlusion avoidance benefit and real-time visual feedback enable precise target selection on smartwatches with an accuracy of 98.9% / 71.5% ~ 100%, which is much more accurate than direct touch.

In summary, the contributions of this work are: (1) presenting Float, a one-handed and touch-free input approach for smartwatches; (2) a participatory design to explore the appropriate motion space for wrist tilt and the *clicking* actions for Float; (3) an interactive implementation of Float only using sensors (motion and PPG) that are available on off-the-shelf smartwatches; (4) empirical studies comparing Float to direct touch.

RELATED WORK

Interaction Techniques on Smartwatches

On smartwatches, the fat finger and occlusion problems are exacerbated by the ultra-small screen. When performing touch interactions, an average of 60% of the area is occluded by the finger over the touchscreen [45]. On a two-dimensional finger touch experiment, error rates were reported ranging from 25% to 66% with target width from 4.8mm to 2.4mm [6]. Smartwatches on the market mostly use enlarged icons and a hierarchical structure for UI design, making many interactions slow and laborious with sequences of swipes and taps [21].

Numerous previous works have explored enhancing the input abilities and input space of smartwatches, such as *Beating Gestures* [28] as well as *WristTap* and *TwoTap* [21]. Ashbrook *et al.* [3] studied inter-target movements for round touchscreen wristwatch. *EdgeTouch* [27], with capacitive sensors around the edge, allows the device screen to remain clearly in view while the finger is touching on the edge. Xiao *et al.* [46] expanded the watch plate into a multi-DOF and mechanical interface. Continuous 2D panning and twist, binary tilt and click can be achieved by interaction on the edge. As for the input space, *SideSight* [7] positions optical sensors along edges of the device allowing virtual touches around the body. Researches also extended the interactive surface to the arm skin through various sensing techniques, such as *Skin Buttons* [23], *SkinWatch* [29] and *SkinTrack* [48].

Researchers have also exploited several factors to improve the pointing accuracy on smartwatches. Oney and Harrison [30] introduced *ZoomBoard*, using iterative zooming to enlarge tiny targets for comfortable sizes. Xia *et al.* [45] presented *NanoStylus*, a finger-mounted tip stylus that enables fast and accurate pointing on a smartwatch with only 16% occlusion. One notable insight is that pointing performance on a smartwatch can be largely improved simply by providing the visual feedback clearly. Ostberg and Matic [31] examined the effects of pre-selection while Yu *et al.* [47] investigated the effects of post-selection feedback for target acquisition. With a little completion time loss, the cursor feedback significantly reduced touch error rates. Float also takes such an advantage, thoroughly.

These previous works above however, all require the other hand to interact. When the other hand is occupied, it will be inconvenient for efficient interaction. We combined tilt and finger actions to overcome this issue.

Tilt Interactions

Tilt has been investigated in a wealth of tasks such as navigating menus [40, 26], 3D rotation [34], text entry [32, 42, 17], mobile touchscreens [9] and authoring system [20]. More surveys can be found in [38]. Crossan [11] *et al.* investigated one-dimensional wrist rotation for mobile interaction with a Fitts' Law analysis. These works including ours follow a basic principle: map the tilt direction and angle to the position of a virtual or latent cursor.

Our present work builds on tilt interaction for selecting tasks in two significant ways. First, previous work primarily focused on binary, discrete or uniaxial control in rectilinear or circular form. For example, a user simply tilts by different inclinations to select an item in a list [26, 40] or simply tilts in discrete (4,8 or 12) directions to acquire an item [10, 14, 32]. Float however, establishes tilt as a more general and continuous 2D pointing method on the smartwatch screen. Second, to our knowledge, our method to calculate the tilt direction and angle is different. The previous typical method is simply computing the magnitude (angle) and arcsin (direction) of pitch and roll [24, 37, 38], which is in fact an approximate and imprecise solution. We present the rigorous calculation based on the intrinsic rotation matrix, which reflects the user's true tilt intention. Although tilt has been testified to be inferior to touch on phone-sized screens, our study showed that after careful calculation and design, tilt enables efficient and much more accurate target selection on a wristwatch than direct touch.

Non-touchscreen and One-Handed Input

Our work is also motivated by bare-handed/non-touch interactions that are designed to mitigate touch problems on the watch face. Abracadabra [15] enables the user to input above and around the device with a high CD gain. The *Gesture Watch* [19] utilizes an array of proximity sensors to detect 10 hand gestures over the device. Shimon *et al.* [2] elicited user-defined [44] gestures for 31 smartwatch tasks. These works still need bimanual interactions, which go against our *one-handed* goal.

To address this issue, one-handed interactions have been explored. *PinchWatch* [22] invokes functions by finger-palm gestures with another camera device attached on body. *ProxiWatch* [25] provides one-handed input by moving the watch towards or away from the body. *GestureWrist* [35] is a specially designed wristband-type device that recognizes hand gestures and forearm movements. Wen *et al.* [41] recognized a set of five fine-motor finger gestures using built-in motion sensors. But the performance would fail when users are walking. More recently, Guo and Paek [14] also take one-handed input on smartwatches as the goal, exploring tilt for wrist-only interactions. The work focused on optimizing the control mechanism with an underlying physics model and presented *ObjectPoint*. They used *tilt-and-persist-to-select* as the confirming action. Although faster than *dwell-to-select*, it was still relatively slow, 1380ms per trial for four targets and 2005ms per trial for 12. In addition, they still placed only one target along a direction (1D pointing circularly). We think this is partly because the confirming action and the tilt-mapping mode are unsuitable for any-location-pointing on smartwatches.

More generally, *DigitSpace* [16] investigated touch performance for one-handed and eyes-free thumb-to-fingers interfaces. Chan *et al.* [8] performed a user-elicitation study eliciting singer-handed microgestures (SHMGs). To recognize hand gestures, previous work integrated external custom-built sensors, such as capacitive sensors [35], force sensitive sensors [12], and strain-gauge sensors [18] onto the watch. In our work, our goal was to recognize finger taps only using sensors (PPG signal from the heart rate monitor and the motion sensor) that are available on common smartwatches. In addition, the algorithm should be accurate and robust in both stationary and mobile contexts.

Target Acquisition Techniques on Screen

Touch benefits from its nature of direct input [1]. However, precise target acquisition is limited on touch screens, especially for small targets. Researchers have developed techniques to address this issue. However, it is generally difficult to apply these techniques to smartwatches.

Earlier pixel-accuracy techniques (*Cross-Keys*, *Precision-Handle* [1], *Dual Finger Selections* [5]) mostly use two fingers and a relatively large extra surface area to possess fine motor control. The callout / popup methods (*Take-Off* [33], *Shift* [39], *TapTap* [36]) avoid occlusion underneath the finger, but meanwhile create new occlusions on other areas. In addition, callout placement on a watch-size screen is much more limited on smartwatches. Back-of-device interaction with a *pseudo-transparency* strategy [43, 4] is also not appropriate for smartwatches due to the form factor.

FLOAT: AN OVERVIEW

Float is a *wrist-to-finger* input interaction, designed as a supplementary technique for touch in hand-busy scenarios. Figure 2 shows a walkthrough of the technique and its coexistence mechanism with touch. Scenario:

(a) **Raise-to-wake:** to interact with the smartwatch, the user raises the arm and turns the wrist so that the watch face faces their line of sight. This *raise-to-wake* action is detected to light up the screen.

(b) **Enabling Float:** The user performs the first in-air finger tap as the *enabling* command to activate the Float interactive mode, after which a (virtual) cursor appears at the center of the watch face.

(c) **Tilt to point:** To select the target, the user tilts the wrist towards a certain *direction* by a certain *angle*. The direction and angle codetermine the cursor location on the watch face.

(d) **Finger tap to select:** Once the correct position is visually verified, a finger tap as the *clicking action* causes a brief highlight and completes the selection.

(e) **Disabling Float:** Any touch event on the screen (indicating that the hand is free) or lowering the arm (indicating that tasks are over) disables Float.

The enabling and disabling commands ensure the separation and compatibility of Float with existing touch interactions on smartwatches, allowing users to explicitly switch between these two input styles.

The main difference between Float and touchscreen-based techniques is that all selecting actions occur within one hand. Float also addresses the *fat finger* problem by allowing the user to always see and verify the pointing position. All UI elements on the screen are visible to users rather than occluded by the wide finger surface or a copied callout. The second difference is that Float uses the polar coordinate system. This requires users to put in different

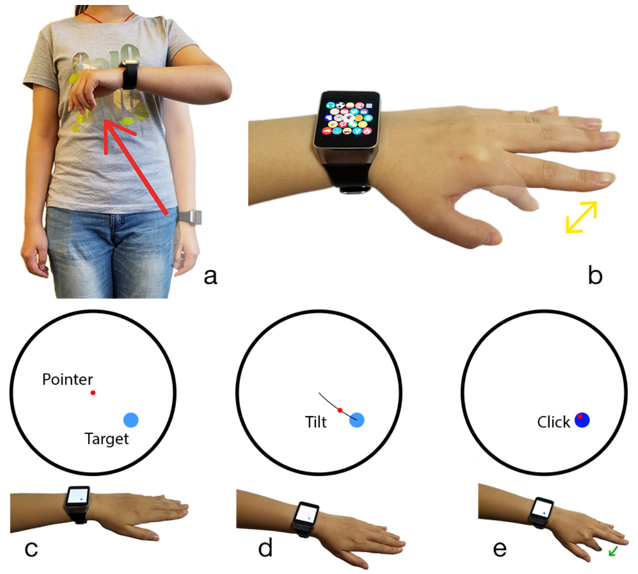


Figure 2. Float input walkthrough. (a) A user raises and turns the wrist to *wake* up the screen; (b) The first in-air finger tap *enables* the Float interaction mode; (c-e) Then the user tilts the wrist to *point* and performs a finger tap to *click*, and continues to point to the next target.

effort selecting targets at different locations, especially along the radial coordinate. The third difference lies in the feedback. Float mainly leverages the proprioception of wrist motion to control.

COORDINATE CALCULATION OF TILT

Float maps wrist tilt to positions in polar coordinates on the two-dimensional watch face. When the watch face tilts to the current orientation from the initial reference, we regard the tilt direction as the cursor 's angular coordinate and map the inclination to the radial coordinate.

Suppose C_0 and C_1 are the initial and the current coordinate system of the watch face respectively. When in C_0, the cursor locates at the center of the watch face, $(0,0)$ on the $xy-plane$. The goal is to calculate the cursor's location after the watch face rotates to C_1. Say (α,β,γ) is the angle change of the intrinsic rotation around the x,y,z axes respectively which transforms C_0 to C_1in the order of $z\rightarrow x\rightarrow y$, and $\boldsymbol{R}$ is its corresponding rotation matrix. So, the unit outward normal vector $\boldsymbol{n_0}=(0,0,1)^T$ of the watch face rotates to (see Eq.1):

$$\boldsymbol{n_1}=(x_1,y_1,z_1)^T=(cos\alpha\ sin\ \beta,-sin\ \alpha, cos\alpha\ cos\beta)^T$$

Projecting $\boldsymbol{n_1}$ onto $xy-plane$ of C_0 gives the tilt direction:

$$Dir=atan2(y_1,x_1)$$

The angle between $\boldsymbol{n_1}$ and $\boldsymbol{n_0}$ gives the inclination:

$$Inc=arccos(z_1)$$

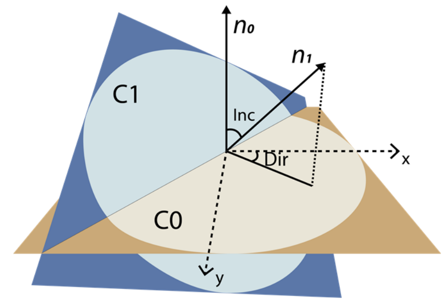


Figure 3. Calculation of the tilt direction and angle when the watch face rotates. A rotation with $\alpha=\beta=30^o$ results in *Inc* = 41.41° and *Dir* = 40.89°.

Such calculation has two characteristics. First, there is a one-to-one correspondence between the wrist orientation and the cursor positions. Any target on the watch face can be reached through appropriate wrist control. Second, Float behaves as users expect. The angular and radial coordinate's mapping accords with users' intuition of locating: The tilt direction of the watch face coincides with the target's angular coordinate; and the farther away the target is located from the center, the more the wrist should tilt. Many previous works [24, 37, 38] simply computed the *magnitude* and *arcsin* of pitch and roll as the tilt direction and angle. Such computation is an approximate method because rotation angles are not equal and have specified orders in intrinsic rotations. For example, if $\alpha=\beta=30^o$, the tilt direction will be 45° using their method. But in fact, it should be ≈ 40.89° (see Figure 3). A rigorous mapping is important for accurate pointing, especially on the ultra-small display of smartwatches.

PARTICIPATORY EXPLORATION AND DESIGN

To better understand how users expect to use Float, we conducted a preliminary study where participants were instructed to explore the appropriate tilt space and elicit *confirming* actions themselves. Study results were used to determine the final design of Float.

Participants

We recruited 12 volunteers (3F/9M, 1 left-handed), ranging in age from 21 to 26 from our university. All of them had experience with ordinary watches and four of them had experience with smartwatches. All participants were given $10 for their participation, which lasted about 40 minutes.

Apparatus

We implemented the tilt calculation described above on a Samsung Galaxy Gear Live smartwatch running Android 5.1. Given the initial and current reference coordinate system, we calculated the intrinsic rotation (γ,α,β) through Android API `getRotationMatrix()` and `getAngleChange()`. The accelerometer and geomagnetic data were collected at

$$\boldsymbol{n_1}=\boldsymbol{R}\cdot\boldsymbol{n_0}=\begin{bmatrix} cos\beta\ cos\gamma+sin\alpha\ sin\beta\ sin\gamma & -cos\beta\ sin\gamma+sin\alpha\ sin\beta\ cos\ \gamma & cos\alpha\ sin\ \beta \\ cos\alpha\ sin\ \gamma & cos\alpha\ cos\gamma & -sin\ \alpha \\ -sin\beta\ cos\gamma+sin\alpha\ cos\beta\ sin\ \gamma & sin\beta\ sin\gamma+sin\alpha\ cos\beta\ cos\ \gamma & cos\alpha\ cos\beta \end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}cos\alpha\ sin\ \beta\\ -sin\ \alpha\\ cos\alpha\ cos\beta\end{bmatrix} \quad (1)$$

a frequency of 180Hz. A filter was applied for each 0.1s sliding window to smooth the white noise. The smartwatch was worn on the wrist of the non-dominant hand. Each participant was asked to stand still and adjust the arm to the most comfortable posture.

Procedure and Design

The study began with the experimenter explaining the principle of Float and the purpose of the study. The whole procedure consisted of two parts.

In the first part, the participant completed a series of discrete *"subjective"* tilting trials. The participant was first asked to explore their own most comfortable position and orientation of the smartwatch. In each trial, the participant first adjusted the watch to the *initial* orientation, where they thought the cursor would be at the center of the watch face. The *calibrate* button was pressed to set and record the initial coordinates. Then a target on the boundary of the circular background was displayed on the watch face. During the trial, a radial cursor reflected the tilting direction. The participant was asked to tilt towards the target taking the cursor as a reference, and then inclined *subjectively* to access the target. The *'subjective'* pointing was based on three conditions: (1) physically comfortable: tilt motion must be easy without arm discomfort; (2) visually comfort-able: the whole UI can be seen and comprehended clearly; (3) they thought subjectively that the inclination angle corresponded to the tilt boundary along this direction. The task consisted of 16 trials, where the targets were arranged uniformly along the boundary of the circular watch face (Figure 4 left). Each participant repeated the task twice.

The second part included two stages. In the first stage, participants were encouraged to design *clicking actions* as much as possible that would cause the "selected" effect based on preference without concern for implementation feasibility in a 'user-defined' [44] manner. After all participants completed the first stage, we summarized all the elicited interactions. Then in the second stage, we explained each interaction to participants and asked them to rate subjective preference scores on a five-point Likert scale for all elicited candidates. Factors they considered included efficiency, comfort and operability, discrete and continuous selection tasks, stationary and mobile contexts and social acceptance. They were advised to wear a smartwatch and evaluate each *clicking action* along with wrist tilt through simulation usage.

Results and Final Design

Tilt Range Determination

For tilt range determination, we collected and analyzed a total of 384 trials from 12 participants. Overall, the appropriate inclination area of wrist-tilt is nearly an oblique ellipse rather than a circle, with an average of 30.15^{o}. A repeated measures ANOVA showed there was a significant effect of tilt directions on the inclination range ($F_{15,165}$=9.83, *p*<.001). The upward direction (around 90^{o}) had the largest inclination range of 39.34^{o}, while the lower left had the smallest range of 23.91^{o}. The lower right range matched the perfect circle most with an average of 29.71^{o} (SD = 0.64).

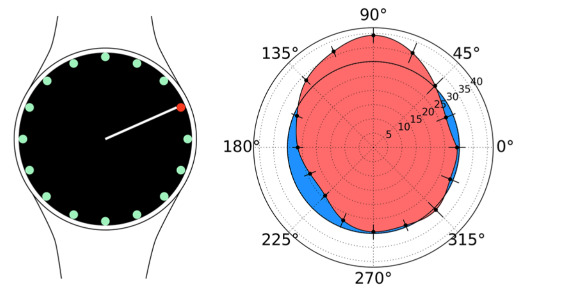


Figure 4. Left: task in exploring the appropriate tilt space; Right: inclination range varies along different directions. The skew pattern was consistent among subjects with variance.

The results suggested that due to the specific wrist structure, participants tended to incline by different angles along different directions. Therefore, to maximize the agreement between users' tilt-motion intention and the displayed cursor position, we scaled the inclination to different ranges as the tilt direction varies. For example, if a user tilted by 20^{o} along the 0^{o} direction, the cursor was located at ($20/29.34$, 0^{o}) = (0.68, 0^{o}) in polar coordinates; but a same 20^{o}-tilt along the 90^{o} direction would result in ($20/39.34$, 90^{o}) = (0.51, 90^{o}). The scaling factor was interpolated by a 1-D cubic spline function along 360^{o}.

Elicited Clicking Action

For user-defined clicking actions, five interaction modalities were elicited (Figure 5). All the participants came up with using *dwell-time* to select while seven out of 12 proposed using *voice control* and *specific wrist motion*. Participants also suggested using *finger actions,* such as an in-air tap and *head / gaze* movement. In the post-rating stage, however, participants rated *finger tap* as the most recognized and appropriate interaction for pointing confirmation, with an average preference score at 4.58. Eleven out of 12 participants voted *finger air-tap* as their favorite option although it had the least elicited times in the first stage.

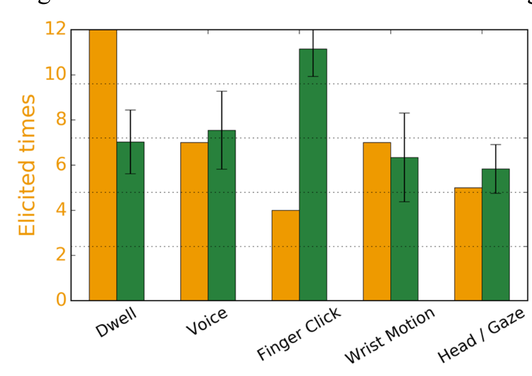


Figure 5. Elicited times and average preference of five user-defined *clicking* interactions in the elicitation study.

Finger tap was recognized as a clear, easy-to-perform and control, and relatively quick method. Participants and pilot tests reported inevitable drawbacks for other options. It is difficult for *Dwell* to be applied on real products because it "*lacks a clear clicking action (P4)*". *Dwell-time* cannot recognize whether a user is holding the wrist to browse the contents or is making selections, and it cannot distinguish between intentional and unintentional dwelling, especially in continuous selecting tasks; Specific wrist motion (e.g. a rapid rotation) would result in cursor's offset and visual interference; *Voice control* is limited due to its inherent problems such as noisy environment and social issues. Not all users are willing to 'speak out' their interactions; Predictably, wink can be faster than finger tap, but suffers from unintended activations from natural eye movements.

Therefore, we chose the *in-air finger tap* as the final design for *clicking*. The next goal was to develop techniques to enable Float interaction, especially to detect *finger taps* accurately and robustly both in stationary and mobile scenarios. We are also expecting to use sensors that have already been embedded in current smartwatches.

TECHNICAL IMPLEMENTATION

The technical implementation of Float includes three parts corresponding to the three transitional actions in Figure 2: raise to wake, tilt and finger tap. We developed the interactive system based on an Android Wear smartwatch. Three types of sensors were used: the motion sensor (accelerometer and gyroscope), position sensor (geomagnetic field) and the heart rate monitor (HRM) sensor.

The motion sensor and the position sensor reflect users' arm and wrist movements, enabling us to recognize the *"raise to wake"* action and compute the cursor position.

The HRM sensor is located on the rear panel of the smartwatch. It detects the PPG signal which measures the amount back of IR/RED light reflected from the blood vessel in the skin. When a user performs finger gestures, significant patterns can be observed from the PPG signal due to the deformation and strain of the wrist skin. So we used the HRM sensor, combined with the motion sensor, to detect users' *clicking action* through *finger tap*. The HRM sensor has been embedded in most commercial smartwatches. But because the raw PPG signal is not open, we decided to attach a standalone HRM sensor (Pulse Sensor [52]) on the rear panel, fixed by a rigid shell by 3D printing (Figure 6.a). The data were processed by an ATmega2560 board at 102 Hz and recognition results were sent to the smartwatch through a Bluetooth module (TI CC2540). Our initial intention was to use the off-the-shelf smartwatch directly if the PPG raw data was available.

Raise-to-Wake Detection

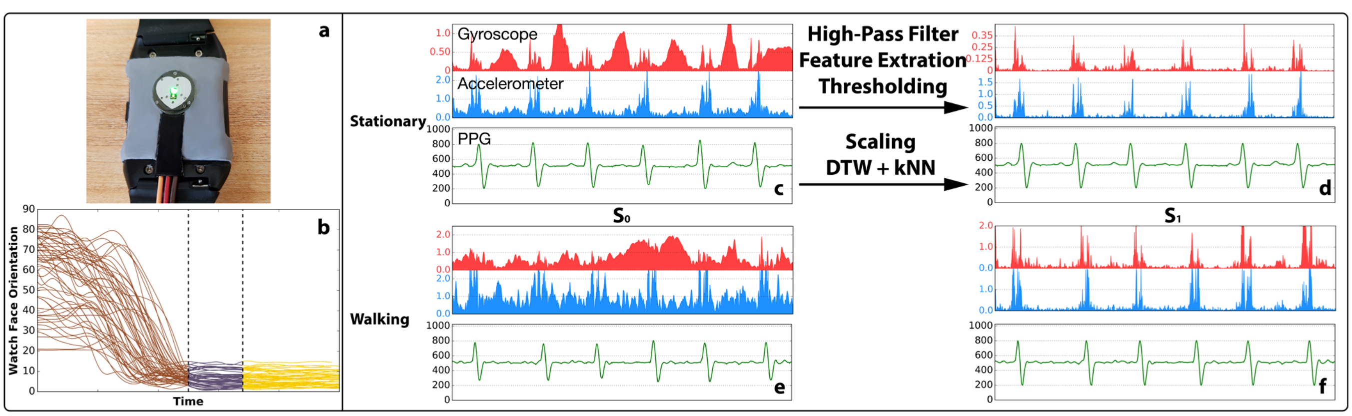
We asked participants to simulate raising arm and turning wrist actions as if they were using the smartwatch in real-life scenarios. Figure 6.b shows the orientation of the watch face against the horizontal ground direction. Within a three-second sliding window, if the inclination was on a declining curve in the first two seconds and stayed stabilized horizontally in the last one second (inclination < 15^{o} and standard deviation < 0.5^{o}), a *raise-to-wake* action was detected.

One-bit Detection of Finger Air-tap

When users perform an *air-tap* action with the finger, joint muscle and skin cause a short but significant change in both motion and PPG signal.

Data Collection

Because smartwatches are usually used in mobile contexts, we collected data in two contexts: stationary and walking (1.2m/s). We recruited 12 participants. For each context, the participant was instructed to wear the smartwatch with comfortable tightness and perform finger taps for five blocks. Each block lasted for 30 seconds with 13 ~ 16 finger taps. In total, 1782 (stationary: 894, walking: 888) trials were collected.



Raise to Wake — Detecting Finger Air-Tap with Motion & PPG signal (the horizontal axis represents time)

Figure 6. Implementation of Float. (a) Pulse Sensor on the rear panel of the smartwatch; (b) Orientation change of the watch face when users raise the watch to light up the screen; (c) ~ (f) When performing finger taps, significant patterns can be observed in the PPG (green) signal and in the accelerometer (blue) and gyroscope (red) sequence after a high-pass Butterworth filter in both stationary and walking contexts.

Detection from Motion Signal

Motion data was mostly full of noise, especially for the walking context (Figure 6.e). Useful information that reflects finger movements in Float needs to be extracted. We first analyzed the distribution in the frequency domain of the accelerometer data after a Fast Fourier Transform (FFT) in these two contexts. It was found that the frequency spectrum of body motion mostly concentrated below 12.4 Hz. So a high-pass Butterworth filter [49] was applied to remove the low-pass (body motion) part, with resonance $=\sqrt{2}$ and cutoff frequency = 12.4 Hz. Figure 6 shows the absolute value of original signal sequence S_0 and the output sequence S_1. The finger air-tap actions resulted in bundles of significant *peaks* in data sequence. To detect such peaks, 6 statistical features were calculated from each T_{tap} sliding window of S_1: *sum, standard deviation and 4 gtX* features. The *gtX* feature is the number of frames whose value is greater than X (set by observation of data) in the window.

The same processing and calculations were performed on the gyroscope data. When and only when all 12 features satisfied certain thresholds (calculated from training data), a *finger tap* action was detected by the motion sensors. However, using only motion data for detection was not reliable. Many other casual but rapid motions, such as jitters would result in similar patterns and cause undesirable false positives, especially in mobile contexts. We combined the PPG signal to improve the robustness.

Detection from Photoplethysmogram Signal

For the PPG sensing data (0~1024), we adopted a Nearest Neighbor Classifier using Dynamic Time Warping (DTW) as a distance measure, which has been shown effective in mining time series data [13]. Specifically, we used the one nearest neighbor classifier on labeled data. Figure 6 shows the PPG signal of users interacting with Float while standing and walking. When performing finger taps, there was a significant pattern – first ascend, then descend, then ascend back. For each sliding window T_{tap}, if *max-min* is greater than a certain threshold (200 in standing context and 350 in walking context), data in the window was normalized between 200 and 800. For training, the average value of each frame of all labeled samples in the training set was computed as the template t. For testing, the distance of DTW between each sliding window s and t decided whether it was a *finger tap* action or not.

Finally, a finger *air-tap* action was detected when and only when both motion and PPG sensor returned true. This constraint was to reduce false positives as much as possible.

Evaluation

The average time of performing an in-air finger tap was 439.7ms (SD=32.1). We evaluated the effectiveness of our detection system through a five-fold cross-validation for each participant. In total, there were zero false positives and nine false negatives out of 894 condition positives for the stationary test while there were seven false positives and 21 false negatives out of 888 condition positives for the walking test. The recall was thus 98.9% and 96.9% respectively. The false discovery rate was 0% and 0.81%. So, it was confirmed that with only PPG and motion sensors, the finger tap actions for Float interaction could be detected accurately.

	Stationary		Walking	
	Finger tap	None	Finger tap	None
Finger tap	885	9	861	20
None	0	/	7	/

Table 1. Confusion matrix of finger air-tap detection: Each column represents the instances in a predicted class while each row represents the instances in an actual class.

Above, we have designed and realized Float to enable one-handed input on smartwatches. We then conducted two user studies to investigate its feasibility and performance.

USER STUDY 1: LOCATION EFFECTS

Float maps the tilt direction and angle of a user's wrist to the corresponding position on a watch face. Based on such a mechanism, we first investigated the effects of target location on the pointing performance.

Task and Procedure

Participants were presented with a series of individual target selection trials. During the study, participants were asked to keep standing. Before each trial, a *calibrate* and a *start* button were displayed at the corners of the watch face, outside the circular background. Participants adjusted their wrist and the smartwatch to their own initial orientation. Participants pressed the *calibrate* button to set the initial coordinates, after which a crosshair cursor appeared at the center. Once the start button was selected by touch, a circular target (r = 0.8 mm) was displayed on the circular background. Participants tilted their wrist to guide the cursor towards and stay inside the target. The trial was completed when the cursor stayed in the target for over 0.4s. Participants were instructed to acquire these targets as accurately and quickly as possible. We did not use finger tap as the selecting action in this study because the focus was on the tilting process.

Design

We used a repeated measures within-participant factorial design. The independent variables were *target Distance* (1/5, 2/5, …, 5/5 of the radius R) and *target Direction* (0^o, 10^o, …, 350^o) on the watch face. Presentation of the *Distances* was counter-balanced across participants. Within each block, the 36 *Directions* were presented in random order. Overall, the experimental design was: 5 $Distances$ × 2 $Blocks$ × 36 $Directions$ = 360 data points per participant.

Participants and Apparatus

The 12 participants who had participated in the pilot study were recruited in this study. All participants were paid $5 for their participation, which lasted about 15 minutes. The experiment was conducted on the Gear Live smartwatch

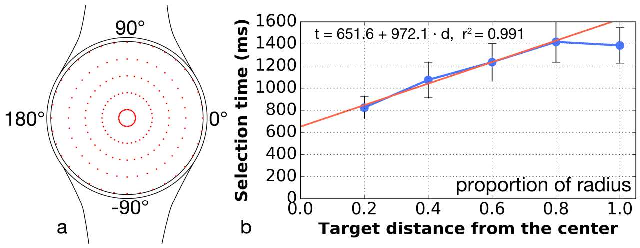


Figure 7. Left: task in Study 1; Right: selection time with different target distances.

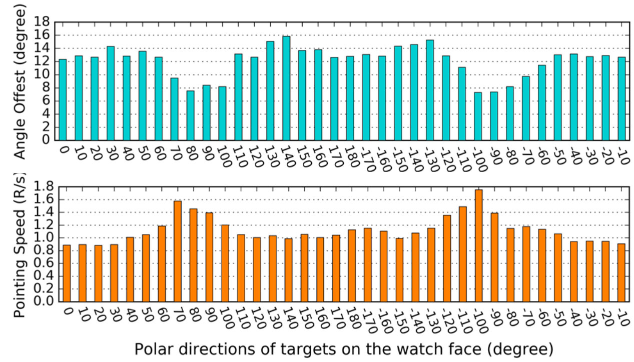


Figure 8. Pointing speed (bottom) and offset angle (top) along different target directions.

described above. It has a 30.5 $\times$30.5mm, 320 $\times$ 320px display with an effective resolution of 10.5px/mm.

Results

Performance with Different Distances

We evaluated the performance by *selection time*, which was measured from the moment the finger was lifted off the start button to the moment the target was selected. A repeated measures ANOVA (same statistical test was used hereinafter) showed that there was a significant effect of *target distance* ($F_{4,44}$=150.60, p<.001) on selection time. Targets at 5/5 distance were faster to acquire than targets at 4/5 because participants could directly tilt by a greater inclination than the mapping limit without much controlling along radial directions.

Considering the first four distances, there was a strong linear correlation between the *selection time* and the *target distance* (r = 0.991). This suggested that participants could guide the cursor to move uniformly by tilting the wrist with the control-display function of linearly mapping from tilt inclination to the radial coordinate on the watch face. The average speed was around 1.03 R/s. The intercept was 651.6ms, containing 251.6ms of reaction time $T_{reaction}$ at the beginning and 400ms of dwell time T_{dwell} at the end.

Performance along Different Directions

For various directions, we measured the performance by the pointing speed, which was defined as

$$v = \frac{Distance}{T_{total} - T_{reaction} - T_{dwell}}$$

Overall, the average pointing speed is 1.12 R/s. We found a significant effect of *target direction* ($F_{35,385}$=5.71, p<.001) on *pointing speed.* There were two distinct speed peaks around +90° and -90°, with 1.58R/s and 1.75R/s respectively, suggesting that participants pointed fastest when tilting the wrist up and down. Another slightly faster direction was around 180° (tilt left), 1.13R/s. Somewhat surprisingly, the pointing speed around the 0° (tilt right) was the lowest, only 0.90 R/s.

We further analyzed the *offset angle* between the actual moving path of the cursor and the ideal straight path, which was defined as the angle between $\boldsymbol{p}_{target} - \boldsymbol{p}_{start}$ and $\boldsymbol{p}_{dmax} - \boldsymbol{p}_{start}$ where $\boldsymbol{p}_{dmax}$ is the point farthest from the ideal straight path. There was also a significant effect of *target direction* ($F_{35,385}$=4.86, p<.001) on *offset angle*. An interesting finding was that there was a strong negative correlation between *pointing speed* and *offset angle* with Pearson's Correlation Coefficient [50] $r = -0.719$, $p < .0001$. Participants guided the cursor with smaller *offset angles* along directions with faster pointing speed. For example, the *offset angle* was only 7.53° along the +80° direction. This suggested that due to the physical structure of the wrist, users have different motion and control abilities along different directions, resulting in different selecting performance.

These results also provide implications for layout and task design of tilt interactions on smartwatches. Overall, important or frequently-accessed targets should be placed as near the center as possible. The exception is that the boundary should be considered before its inner neighboring region. For targets with the same distance from the center, we should give priority to angles around +90°, -90° and 180°.

USER STUDY 2: CONTINUOUS POINTING

In this study, we evaluated the overall performance of Float on continuous pointing tasks.

Task and Procedure

Participants were presented with a series of sequential target selection trials. A total of nine target sizes and two usage contexts were tested. Participants were instructed to follow the Float scenario and acquire these targets as accurately and quickly as possible.

At the beginning of each block, participants raised the arm to *wake up* the task. Then round targets were presented con-

Figure 9. Study 2 was conducted in both standing (left) and walking (right) contexts.

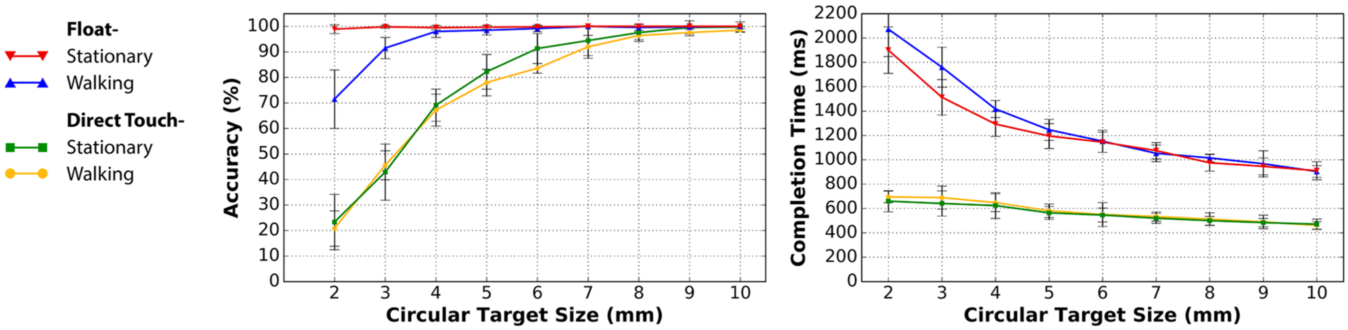


Figure 10. Mean accuracy and completion time of continuous pointing for *Technique* and *Context* over different target sizes.

secutively on the circular background of the watch face. Once a target appeared, participants guided a cursor in crosshair shape towards the target by tilting the wrist. When the cursor was into the target, participants performed an in-air finger tap to confirm and complete the selection. The cursor location right before the finger tap was set as the *clicking* position. Participants advanced to the next trial immediately regardless of error.

Because there is still a lack of baseline techniques for one-handed smartwatch interaction in research, in this study we compared Float with direct touch which is widely used on smartwatches. It is worth noting that Float enables input with only one hand, which is the basis of work. The exper-iment of direct touch was more concerned with providing a visual reference on performance than merely comparison.

Design

A repeated measures within-participant factorial design was used. The independent variables were *Technique* (*Float* and *Touch*), interaction *Context* (*Stationary*, *Walking*), and *Target Size* (diameter: 2, 3, 4, 5, 6, 7, 8, 9, 10 mm). Presentation of *Technique* and *Context* was counterbalanced across participants. The target size as small as 2~5mm was to measure the precision and ability of Float. We tested the performance while *Walking* because smartwatches are usually used in wild and mobile scenarios.

The 9 *Target Sizes* were presented in random order for each participant. For each condition, participants completed 1 practice block and 5 testing blocks. Each block consisted of 10 targets whose positions were randomly generated across the watch face. The average distance between targets was around 13.8mm, 128/45π of the screen radius. The targets were randomly placed to simulate actual usage and test its average performance. We did not follow the standard Fitts' Law design in a circular form because unlike touch screen, Float is location-sensitive. Targets may appear anywhere and tilt paths mostly do not cross the center.

In summary, the experimental design was: 2 *Techniques* × 2 *Contexts* × 9 *Target Sizes* × 5 *Blocks* × 10 *Trials* = 1800 data points per participant.

Participants and Apparatus

The 12 participants who had participated in the data collection of the implementation section were again recruited for this study. Finger taps were detected with the method above. For the *Walking* condition, participants were asked to walk on a treadmill at the speed of 1.2m/s [51] while performing selecting tasks. All participants were paid $20 for their participation, which lasted about 90 minutes.

Results

Accuracy

In *stationary* context, participants achieved almost 100% of accuracy on all target sizes from 2mm to 10mm with Float. This benefited from its clear visual feedback and participants could directly verify whether the cursor was inside the target or not. In addition, a high pointing accuracy on 2mm (98.9%) suggested that participants could hold the cursor steadily by controlling the wrist.

We found a significant effect of *Context* ($F_{1,11}$=42.42, p<.001) on the pointing accuracy of Float. During *walking*, there was an obvious decrease on performance on *size* 2mm and 3mm with an accuracy of 71.5% and 91.5%, respectively. This was because during walking, the body movement itself had effects on the wrist movement and the accelerometer of the motion sensor, causing the cursor fluctuation on the watch face. With *size* no less than 4mm, however, participants were able to perform high-precision (> 98.0%) target selection with Float.

A repeated measures ANOVA showed that there was a significant main effect of *Technique* on *accuracy* in both *stationary* ($F_{1,11}$=68.29, p<.001) and *walking* contexts ($F_{1,11}$=598.77, p<.001). Figure 10 also demonstrates that Float was more accurate than direction touch over all target *sizes*, especially for smaller targets from 2mm (85.21%% vs 22.07%) to 6mm (99.50% vs 87.47%). Touch can only reach an accuracy of 70% at *size* 4mm and 90% at 6mm.

Completion Time

The completion time of Direct Touch was significantly ($F_{1,11}$=598.77, p<.001, 1240ms at *size* 2mm and 439ms at 10mm) smaller than Float, which benefited from its nature of direct input. For Float, there was a significant effect of *Target Size* ($F_{8,88}$=344.62, p<.001) on completion time. In *stationary* context, participants spent an average of 1900ms at *size* 2mm to 910ms at *size* 10mm. The completion time decreased largely (607ms) from *size* 2mm to *size* 4mm and then gradually to *size* 10mm (373ms).

There was also a significant effect of *Context* ($F_{1,11}$=11.36, p<.01) on the accuracy of Float. Participants spent more time (51~172ms) on selecting smaller (2~5mm) targets during *walking* than staying *stationary* due to the unstable cursor. On larger (6~10mm) targets, the completion time showed no significant difference.

DISCUSSION AND LIMITATIONS

We wanted to explore if the combination of wrist tilt and finger tap could improve one-handed and touch-free input on smartwatches. Our experimental results support our expectation. The occlusion avoidance benefit and real-time visual feedback of Float enable precise target selection on smartwatches. In the walking context, the accuracy performance only reduced for ultra-small targets (< 4mm). To complete a selection, participants spent an average of ~1s to ~2s on targets from large (10mm) to small (2mm). The completion time of Float doubles or triples the time of touch due to the *pointing* and *clicking* processes.

Lastly, we revisit part of our work and discuss the lessons we learned and the limitations of our approach.

Screen Visibility

In Float, the inherent issue with wrist tilt is that it changes the viewing angle and the screen visibility may be of concern. The visual ability on the tilted screen is indeed unevenly distributed. To ensure its feasibility, we have investigated the appropriate boundaries for the wrist tilt in the elicitation study so that users could see and comprehend well the entire contents over the watch face. Figure 11 illustrates the visual perception in practical experience with Float when the user tilted the watch face by the maximum angle along three typical directions. Even small texts of 2mm height near the boundary could be seen very clearly.

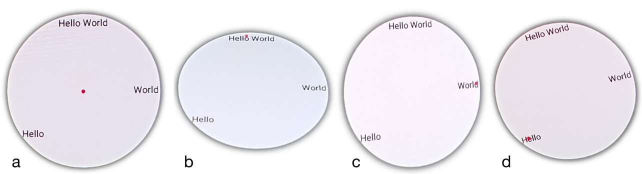


Figure 11. Visual perception when using Float from the first perspective. The red point represents the cursor. Tilt angles are (a) 0^o at the initial phase, (b) ≈40^o when tilting upward, (c) ≈30^o when tilting to the right, and (d) ≈25^o when tilting to the lower left corresponding to the tilt ranges in Figure 4.

Another visual limitation is that the cursor might occlude contents when interacting with Float. We think the appearance of the cursor should be content-dependent. For example, when the interface is organized as clicking objects (e.g. buttons, list items), it can hide the cursor and only highlight the target background; when the interface is for setting a specific position or value, the cursor should appear.

Motion and Perception

The non-linear input mapping in the elicitation study and the pointing performance in study 1 reflect different motion abilities and perception of wrist tilt along directions (Figure 4, Figure 8). A significant phenomenon from observation is the subtleness of the body movement. Up/down tilts are small and discrete motions, while left/right tilts are much larger because the forearm, the elbow and sometimes even the upper arm need to jointly move. This also partly explains the unevenly distributed performance on the watch face of Float. Overall, tilting up/down achieved both faster and more stable pointing performance than that of tilting to the left/right. Participants also expressed their subjective feedback on this: *"I need to raise or lower my elbow to point to some areas." (P2)* and *"The physical cost is different but does not affect the user experience much" (P4)*. We will analyze the arm motion and examine its influence on feasibility in actual scenarios in future work.

Finger Tap and User Evaluation

The presented user evaluation is limited because in the experiment, we only tested the performance of finger tap as the selecting action, which was derived from a user-elicitation study with usability analysis. Although finger tap performs as hypothesized, we believe other elicited input styles or their variants also have their own merits. Future work will include a statistical comparison of all mentioned techniques in Figure 5 to understand the usability and advantages of each approach in real-world practice.

Application Examples

In study 2, we have confirmed the accuracy of Float on 2mm-10mm target sizes which are smaller than common targets on a touchscreen. This first proves that Float can be directly applied to current UIs on smartwatches. Meanwhile, Float can enable tasks that require precise pointing which are difficult to achieve on smartwatches due to the small form factor and the fat finger problem. Application examples may include specifying a location on a map, selecting color in a color picker, inserting the cursor in the text and inputting characters with a mini keyboard (< 3mm per key).

CONCLUSION

We have explored input interactions on smartwatches when the other hand is occupied. With Float, a user tilts the wrist to *point* and performs an in-air finger tap to *click*. We have conducted a participatory study to explore the appropriate wrist tilt space and let users define the *clicking* interaction. To realize Float, we only use sensors that are commercially embedded on smartwatches. We first present how to map the wrist tilt movement to the corresponding positions on the watch face. Then we recognize finger taps robustly using the PPG signal and the motion sensor. Results from two user studies have confirmed the feasibility of Float which provides quick and high-precision target selection.

ACKNOWLEDGEMENT

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001402, the Natural Science Foundation of China under Grant No. 61272230 and No. 61572276, Tsinghua University Research Funding No. 20151080408.

REFERENCES

1. Pär-Anders Albinsson and Shumin Zhai. 2003. High precision touch screen interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '03), 105-112. DOI=http://dx.doi.org/10.1145/642611.642631
2. Shaikh Shawon Arefin Shimon, Courtney Lutton, Zichun Xu, Sarah Morrison-Smith, Christina Boucher and Jaime Ruiz. 2016. Exploring Non-touchscreen Gestures for Smartwatches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 3822-3833. DOI=http://dx.doi.org/10.1145/2858036.2858385
3. Daniel Ashbrook, Kent Lyons and Thad Starner. 2008. An investigation into round touchscreen wristwatch interaction. In *Proceedings of the international conference on Human computer interaction with mobile devices and services* (MobileHCI '08), 311-314. DOI=http://dx.doi.org/10.1145/1409240.1409276
4. Patrick Baudisch and Gerry Chu. 2009. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1923-1932. DOI=http://dx.doi.org/10.1145/1518701.1518995
5. Hrvoje Benko, Andrew D. Wilson and Patrick Baudisch. 2006. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), 1263-1272. DOI=http://dx.doi.org/10.1145/1124772.1124963.
6. Xiaojun Bi, Yang Li and Shumin Zhai. 2013. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1363-1372. DOI=http://dx.doi.org/10.1145/2470654.2466180
7. Alex Butler, Shahram Izadi and Steve Hodges. 2008. SideSight: multi-"touch" interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST '08), 201-204. DOI=http://dx.doi.org/10.1145/1449715.1449746
8. Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang and Frank Maurer. 2016. User Elicitation on Single-hand Microgestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 3403-3414. DOI=http://dx.doi.org/10.1145/2858036.2858589
9. Youli Chang, Sehi L'Yi, Kyle Koh and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 1499-1508. DOI=http://dx.doi.org/10.1145/2702123.2702425
10. Andrew Crossan and Roderick Murray-Smith. 2004. Variability in wrist-tilt accelerometer based gesture interfaces. In *Proceedings of the International Conference on Mobile Human-Computer Interaction* (MobileHCI '04), 144-155.
11. Andrew Crossan, John Williamson, Stephen Brewster and Rod Murray-Smith. 2008. Wrist rotation for interaction in mobile contexts. In *Proceedings of the international conference on Human computer interaction with mobile devices and services* (MobileHCI '08), 435-438. DOI=http://dx.doi.org/10.1145/1409240.1409307
12. Artem Dementyev and Joseph A. Paradiso. 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), 161-166. DOI=http://doi.acm.org/10.1145/2642918.2647396
13. Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang and Eamonn Keogh. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. In *Proceedings of the VLDB Endowment 1*, 2: 1542-1552. DOI=http://dx.doi.org/10.14778/1454159.1454226
14. Anhong Guo and Tim Paek. 2016. Exploring tilt for no-touch, wrist-only interactions on smartwatches. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16), 17-28. DOI=http://dx.doi.org/10.1145/2935334.2935345
15. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 121-124. DOI=http://dx.doi.org/10.1145/1622176.1622199
16. Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 1526-1537. DOI=http://dx.doi.org/10.1145/2858036.2858483
17. Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani and Sriram Subramanian. 2010. GesText: accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 2173-2182. DOI=http://dx.doi.org/10.1145/1753326.1753655

18. Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. WristWhirl: One-handed Continuous Smartwatch Input using Wrist Gestures. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16), 861-872. DOI=https://doi.org/10.1145/2984511.2984563
19. Jungsoo Kim, Jiasheng He, Kent Lyons and Thad Starner. 2007. The gesture watch: A wireless contact-free gesture based wrist interface. In *Proceedings of the International Symposium on Wearable Computers* (ISWC '07), 1-8. DOI=http://dx.doi.org/10.1109/ISWC.2007.4373770
20. Ju-Whan Kim, Han-Jong Kim and Tek-Jin Nam. 2016. M. Gesture: An Acceleration-Based Gesture Authoring System on Multiple Handheld and Wearable Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 2307-2318. DOI=http://dx.doi.org/10.1145/2858036.2858358
21. Benjamin Lafreniere, Carl Gutwin, Andy Cockburn and Tovi Grossman. 2016. Faster Command Selection on Touchscreen Watches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 4663-4674. DOI=http://dx.doi.org/10.1145/2858036.2858166
22. Loclair C, Gustafson S, Baudisch P. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '10)
23. Gierad Laput, Robert Xiao, Xiang 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), 389-394. DOI=http://dx.doi.org/10.1145/2642918.2647356
24. I. Scott MacKenzie and Robert J. Teather. 2012. FittsTilt: the application of Fitts' law to tilt-based interaction. In *Proceedings of the Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (NordiCHI '12), 568-577. DOI=http://dx.doi.org/10.1145/2399016.2399103
25. Florian Müller, Sebastian Günther, Niloofar Dezfuli, Mohammadreza Khalilbeigi and Max Mühlhäuser. 2016. ProxiWatch: Enhancing Smartwatch Interaction through Proximity-based Hand Input. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '16), 2617-2624. DOI=http://dx.doi.org/10.1145/2851581.2892450
26. Ian Oakley and Sile O'Modhrain. 2005. Tilt to scroll: evaluating a motion based vibrotactile mobile interface. In First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference (HAPTICS '05), 40-49. DOI=http://dx.doi.org/10.1109/WHC.2005.138
27. 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), 169-178. DOI=http://dx.doi.org/10.1145/2556288.2557138
28. Ian Oakley, Doyoung Lee, M.D Rasel Islam and Augusto Esteves. 2015. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 1237-1246. DOI=http://dx.doi.org/10.1145/2702123.2702226
29. Masa Ogata and Michita Imai. 2015. SkinWatch: skin gesture interaction for smart watch. In *Proceedings of the Augmented Human International Conference* (AH '15), 21-24. DOI=http://dx.doi.org/10.1145/2735711.2735830
30. Stephen Oney, Chris Harrison, Amy Ogan and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2799-2802. DOI=http://dx.doi.org/10.1145/2470654.2481387
31. Anna Ostberg and Nada Matic. 2015. Hover cursor: Improving touchscreen acquisition of small targets with hover-enabled pre-selection. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '15), 1723-1728. DOI=http://dx.doi.org/10.1145/2702613.2732903
32. Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello and Roy Want. 2002. TiltType: accelerometer-supported text entry for very small devices. In *Proceedings of the 15th annual ACM symposium on User interface software and technology* (UIST '02), 201-204. DOI=http://dx.doi.org/10.1145/571985.572013
33. Richard L. Potter, Linda J. Weldon and Ben Shneiderman. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '88), 27-32. DOI=http://dx.doi.org/10.1145/57167.57171
34. Jun Rekimoto. 1996. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology* (UIST '96), 167-168. DOI=http://dx.doi.org/10.1145/237091.237115
35. Jun Rekimoto. 2001. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices.

In *Proceedings of the International Symposium on Wearable Computers* (ISWC '01), 21-27.

36. Anne Roudaut, Stéphane Huot and Eric Lecolinet. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces* (AVI '08), 146-153. DOI=http://dx.doi.org/10.1145/1385569.1385594
37. Keigo Shima, Kazusa Onishi, Ryosuke Takada, Takuya Adachi, Buntarou Shizuki and Jiro Tanaka. 2016. Investigating Accuracy of Tilting Operation on Wrist-worn Devices with Touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 2705-2711. DOI=http://dx.doi.org/10.1145/2851581.2892377
38. Robert J. Teather and I. Scott MacKenzie. 2014. Position vs. velocity control for tilt-based interaction. In *Proceedings of Graphics Interface* (GI '14), 51-58.
39. Daniel Vogel, and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing System* (CHI '07), 657-666. DOI=http://dx.doi.org/10.1145/1240624.1240727
40. Lars Weberg, Torbjörn Brange and Åsa Wendelbo Hansson. 2001. A piece of butter on the PDA display. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '01), 435-436. DOI=http://dx.doi.org/10.1145/1240624.1240727
41. Hongyi Wen, Julian Ramos Rojas and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 3847-3851. DOI=http://dx.doi.org/10.1145/2858036.2858466
42. Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03), 81-90. DOI=http://dx.doi.org/10.1145/964696.964705
43. Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell and Chia Shen. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (UIST '07), 269-278. DOI=http://dx.doi.org/10.1145/1294211.1294259
44. Jacob O. Wobbrock, Meredith Ringel Morris and Andrew D. Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1083-1092. DOI=http://dx.doi.org/10.1145/1518701.1518866
45. 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 and Technology* (UIST '15), 447-456. DOI=http://dx.doi.org/10.1145/2807442.2807500
46. 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) 193-196. DOI=http://dx.doi.org/10.1145/2556288.2557017
47. Chun Yu, Hongyi Wen, Wei Xiong, Xiaojun Bi and Yuanchun Shi. 2016. Investigating Effects of Post-Selection Feedback for Acquiring Ultra-Small Targets on Touchscreen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 4699-4710. DOI=http://dx.doi.org/10.1145/2858036.2858593
48. Yang Zhang, Junhan Zhou, Gierad Laput and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 1491-1503. DOI=http://dx.doi.org/10.1145/2858036.2858082
49. https://en.wikipedia.org/wiki/Butterworth_filter
50. https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient
51. https://en.wikipedia.org/wiki/Preferred_walking_speed
52. www.pulsesensor.com