TriTap: Identifying Finger Touches on Smartwatches

Hyunjae Gil[1], DoYoung Lee[1], Seunggyu Im[2], Ian Oakley[1]
[1]Dept. of Human Factors Engineering, UNIST, Ulsan, Korea
[2]Jacobs University, Bremen, Germany
{hyunjaegil, dylee.issac, seunggyu.im, ian.r.oakley}@gmail.com

ABSTRACT
The small screens of smartwatches provide limited space for input tasks. Finger identification is a promising technique to address this problem by associating different functions with different fingers. However, current technologies for finger identification are unavailable or unsuitable for smartwatches. To address this problem, this paper observes that normal smartwatch use takes places with a relatively static pose between the two hands. In this situation, we argue that the touch and angle profiles generated by different fingers on a standard smartwatch touch screen will differ sufficiently to support reliable identification. The viability of this idea is explored in two studies that capture touches in natural and exaggerated poses during tapping and swiping tasks. Machine learning models report accuracies of up to 93% and 98% respectively, figures that are sufficient for many common interaction tasks. Furthermore, the exaggerated poses show modest costs (in terms of time/errors) compared to the natural touches. We conclude by presenting examples and discussing how interaction designs using finger identification can be adapted to the smartwatch form factor.

Author Keywords
Smartwatch; Finger Identification; Touch Contact Profile

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—Input Devices and Strategies (e.g., mouse, touchscreen)

INTRODUCTION
Interaction with wearable devices such as smartwatches is a highly constrained experience. Small screens and touch surfaces provide few opportunities to create powerful, expressive interfaces using the conventional tap and swipe input popularized on larger devices such as smartphones. This problem is both fundamental, in that users want small, useful, discrete wearable devices, and interesting, in that it raises substantial new challenges for researchers and designers working in the space. Indeed, as more powerful and mature wearables come to market, increasing attention in the HCI community is being devoted to interacting with them in non-traditional ways, such as using their on-board inertial sensors to detect physical movements of the watch [30, 32] or gestural movements of the hand and fingers [28]. Commercial work in this area includes the Apple Watch (www.apple.com/watch) and its use of pressure input [23] and its crown, a novel physical controller.

While these approaches show considerable promise, the touch screens of smart watches remain powerful high fidelity sensors with a direct connection to the primary display surfaces of the device. As such, researchers are also exploring how to maximize the value of touch input on tiny screens through techniques such as diverse as tapping gestures [17], multi-touch menu systems [11] and inferring touch properties such as finger angle [29]. Within this space, one promising technique for increasing the expressiveness of touch input is finger identification [21]. At heart, this is a simple idea: if a system can process screen touches to disambiguate which of a user's fingers is responsible for a touch, then different functions or operations can be assigned to each finger. A range of prior work on this topic has discussed the interaction and application scenarios enabled by this technique in contexts as diverse as physical buttons [23], tablets [4] and tabletop computers [3].

However, while technologies to achieve finger identification have been achieved in fixed [8] or large format devices [9], they remain challenging to implement in small or mobile devices. In fact, most current work on this topic on wearables or mobiles simply instruments the touching fingers [5] to support system development or assumes instructions as to which finger should be used will be faithfully followed during studies [21]. While these approaches are effective for pursuing purely application design or empirical goals, they are also somewhat impractical – it is unlikely that real users will commit to wearing sensors or markers on their fingers simply to interact with another wearable device. Existing research to classify touches on small devices does exist, such as Harrison *et al.*'s [7] use of touch impact sounds to identify hand regions such as the fingertip, pad, nail or knuckle. However, little work has examined how we might use the properties of a touch to distinguish between fingers, rather than finger regions, on a small wearable device.

CHI 2017, May 06-11, 2017, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/3025453.3025561

This paper aims to fill this gap by building on recent research showing it is possible to extract finger angles from the raw touch image generated by a standard capacitive touch screen [29]. Within this space, the contributions of this paper are threefold. Firstly, it proposes the idea that, in the constrained input poses available on a smartwatch (i.e. fixed to the wrist and approachable by the touching hand from a very limited range of angles) the touch profiles and angles generated by a user's fingers may by sufficiently distinctive to support reliable finger identification. Secondly, it contributes an empirical investigation of the validity of this idea: two studies of common smartwatch input techniques capture touch in both natural input conditions and those in which participants are instructed to exaggerate the angles of their finger touches. Our analysis describes the touches and discusses the efficiency and accuracy of user input and the recognition rate for finger identification in these scenarios. Thirdly, this paper contributes a discussion of how finger identification could be realistically implemented in current wearables and the types of interaction it could support. This discussion is showcased with application examples including two- and three-finger keyboard designs (and limited validations) and *tricons,* smartwatch application icons customized for use with finger identification technology.

RELATED WORK

Smartwatches and other wearable devices are powerful computational tools packaged with highly limited input and output capabilities. Authors are responding to the design opportunity this represents by proposing techniques to enhance and enrich interactions. The scope is broad, spanning topics as diverse as sensing input [27] or producing output on the skin [12], integrating touch sensitive surfaces into alternative parts of a device such as the strap [19] or edge [16] and utilizing body sensing techniques such as EMG [22] or tomography [31] to infer user actions. In contrast, finger identification has received limited attention. Gupta and Balakrishnan provide the most thorough exploration of the potential of this technique on smartwatches in their study of DualKey [5], a keyboard in which letters are clustered in horizontal pairs. Tapping on a pair with the index finger selects the leftmost letter, while tapping with the middle finger selects the rightmost letter. This effectively doubles target sizes and Gupta's comprehensive study highlights the resulting performance benefits. However, their system remains a lab prototype as it relies on a cumbersome finger mounted distance sensor to disambiguate touches by one finger from those by the other.

Finger identification has received more attention on larger platforms. Attracted by the simplicity of the technique there is a relatively large body of research that can be broadly categorized into technical systems for recognizing fingers or user centered investigations into the design of [21], or performance with [4], the technique. In terms of designs, most proposals are variations on the idea that specific fingers can be used to access different functions. For example, one might trigger a context menu [7], or provide easy access to commonplace commands such as cut, copy and paste. In terms of evaluations, studies have catalogued user performance with different fingers during tapping and dragging tasks on, for example, tablet computers [4]. Roy et al. [21] provide a laudably comprehensive review this literature. However, we note that while this work represents a substantial design and empirical resource relating to interaction with systems capable of identifying touching fingers, few articles have considered the specific form factor of smartwatches: small touch screens mounted on one wrist. Data and design guidance for larger devices likely needs updating or refinement for small screens.

Techniques for identifying the finger making a touch come in several forms. In the simplest, the finger is instrumented with a marker [15, 25] or sensor [5]. Another approach has been to infer individual finger locations in the context of multiple simultaneous touches based on the physical constraints imposed by the shape of the hand [3]. This technique has limited potential on the small screens of smartwatches. The Beats system [17], for example, accepts a pair of simultaneous touches on a watch and associates the left-most one with the index finger and the rightmost one with the middle finger to form sequences of tap gestures. However, all further position information is discarded, limiting the scope of the system for general purpose interaction. Other approaches for identifying fingers involve sophisticated touch surfaces capable of, for example, detecting the finger prints of the touching fingers [8, 24], or rely on advanced visual tracking systems positioned either under [2] or over a screen. As Gupta and Balakrishnan [5] note, none of these techniques is available or suitable for use with the current touchscreen technology available in smartwatches.

To address this technological lack, this paper explores whether the data reported by a standard capacitive touch screen in terms of finger contact area and finger angle [29] is sufficient to identify the touching finger. Many prior authors have recognized the value that can be gained by using finger contact area as an input modality. For example, Wang *et al*. [26] discuss the elliptical nature of touches on a tabletop and how this can be used to create interaction techniques such as ray based pointing. Other authors have applied these ideas to mobiles. Boring *et al.* [1] discuss how changes to the profile of a thumb touch on a phone can be used to shift between interface modes while Rogers *et al.* [20] show that tracking finger angle can improve pointing performance and create interfaces that automatically adjust for finger occlusions or scroll through menu options based on pitch. Recently Xiao *et al.* [29] discuss the accuracy with which single finger orientations can be inferred using the raw data from a standard watch or phone touch screen. The goal of this paper is to build on these findings and ideas to create a robust finger identification system that operates with data from currently available touch screen technology.

PERFORMANCE STUDIES

The main goal of these studies was to explore the viability of using touch contact area to recognize fingers during interaction with a smartwatch. We considered three fingers: *thumb*, *index* and *middle*, as prior research has suggested that ring and pinky fingers are rarely deployed by users in touch input tasks on smartwatches [18] and, indeed, perform relatively poorly in situations where they are used [4]. In line with prior work, we conducted two studies to cover two common forms of touch screen input: *tapping*, or selecting targets by touching the screen, and *swiping*, or making rapid stroke gestures in cardinal directions.

We also considered two input conditions: *natural* and *exaggerated*. In the natural conditions, participants performed input tasks in any way they were comfortable. In the exaggerated conditions, participants were instructed to make input by using the left side of their thumbs, the tip of their index fingers and the right side of the middle fingers. These poses are shown in Figure 1 (on our phone based study prototype) and were selected to maximize the distinctiveness of the touch contact profiles of the different fingers while also remaining convenient and comfortable in the context of smartwatch use. The goal of including both natural and exaggerated conditions was twofold. First, it enabled us to investigate the feasibility of inferring the touching finger during natural interaction. Second, it allowed us to characterize both the benefits of requiring participants to use a specific pose for input (in terms of finger recognition performance) and the costs of doing so (in terms of time, errors or comfort and workload). The following sections describe these studies in detail.

System

A wide variety of mechanisms have been deployed to gain access to detailed information about touch contact regions. These include using the standard reporting methods in the operating system [1], using specialized hardware such as fingerprint scanners [8], constructing bespoke sensor grids [18] and modifying the touch drivers on existing smart devices [10, 29]. We followed this latter approach as it provides high fidelity data while relying on commonplace and relatively high performance sensing hardware. Using a commodity device also supports our objective of exploring whether reliable finger identification is possible using current technology. However, as the main goals of this work are empirical, we opted to simplify the development process (in terms of better documentation and easier access to features such as network connectivity and storage) by using a region of an Android smartphone as a surrogate for a smartwatch. We note this is a common approach [e.g. 13] and that touch sensors used in both classes of device are reported to be very similar [29].

As with prior authors [10], we modified the Android kernel to poll the touch screen driver for the touch image – the raw sensor data recorded by each capacitive electrode. This data varies with the proximity of each sensor to a touching finger to form a greyscale intensity image that captures the finger-screen contact area and, to a lesser extent, the finger regions directly above the screen [27]. Our implementation ran on a Google Nexus 5 Smartphone and captured 16-bit touch intensity data in an eight by eight sensor grid covering an area 33mm square in the top right corner of the phone at 33Hz. During the initial studies, we processed this data using flood fill based blob detection and ellipse fitting [18] to derive a centroid for each screen touch. The modified kernel source files and example applications are available for download (https://github.com/UNIST-Interactions/tritap). Figure 2 shows the system in operation.

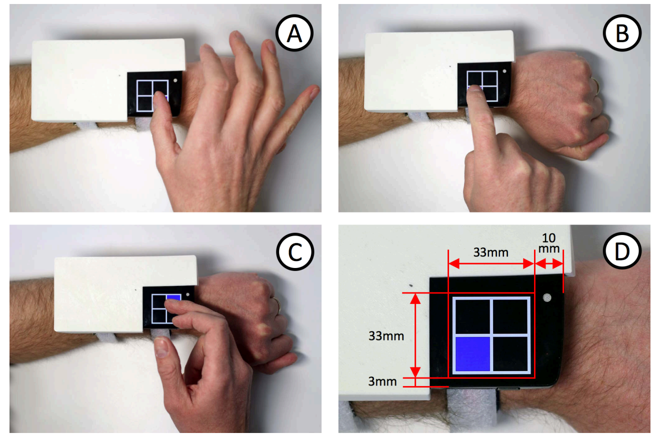

Figure 1. Experimental setup: wrist mounted phone showing four target tapping condition and exaggerated touches with thumb (A), index (B) and middle (C) fingers. Close up including annotations of sensor region and bevel sizes (D).

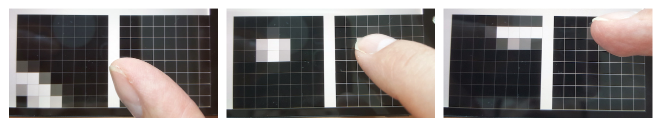

Figure 2. Three touches on the 8x8 33mm square touch sensor grid on the Nexus 5 phone. The thumb (left), index (center) and middle (right) fingers are touching the screen in the exaggerated poses. The left region of each image shows the sensor data generated by the touch on the right.

Methods

Twenty participants (mean age 22, nine female) completed the tapping study while nine (mean age 21, five female) completed the swiping study. They were recruited from the local student body via online methods and word of mouth and received ~10 USD as compensation. We screened for right-handedness. Over both studies, participants rated themselves as familiar with smartphones (mean 4.5/5) and touchscreens (mean 4.6/5) but not wearables (mean 1.4/5). No participant completed both studies.

In both studies participants rested their left arms comfortably on a desk top in front of them and wore the smartphone strapped along their wrist using two watchbands attached to the back of the phone with Velcro. The top right corner of the phone was adjusted to be in a typical location for a watch: center of the wrist, just back from the hand. It also had the smallest possible bevels at its base (~3mm) and right edge (~10mm). A 3D printed cover obscured the rest of the phone and served to indicate the study touch area to participants. Figure 1 shows this setup, including annotations showing bevel sizes. All content in the study was shown in the 33mm square region used for data capture and each trial took the same form. First participants tapped the screen to start. A hand graphic highlighting which finger to use was then presented for 1000ms, followed by the experimental trial. This took the form of a grid of targets in the tapping study and a single target and direction in the swiping study. The instructions are illustrated in Figure 3. As with prior work [21], we did not independently verify if participants used the requested finger in each trial. The task is simple and prior work suggests the compliance rate will be very high.

In both studies all participants completed natural input conditions prior to exaggerated conditions. This unbalanced repeated measures design ensured that the instructions given about poses in the exaggerated conditions did not impact the touches recorded in the natural conditions. Both studies were composed of sequences of identical trial blocks. In the tapping study three were three blocks, the first of which was treated as practice and discarded. In the shorter swiping study, there were five blocks, the first two of which were treated as practice and discarded. Within each block in both studies trials were delivered in a random order and participants were required to repeat error trials.

In the tapping study, two target sizes were used: a 2x2 grid and a 3x3 grid, corresponding to targets of 13mm and 8.25mm square. There was a 3mm border around the targets and an inter-target spacing of ~1mm. Each block of trials was composed of a set of trials in the 2x2 grid and a set of trials in the 3x3 grid. The order of the sets was balanced among participants. In both sets, participants were required to complete six trials per target, three with each finger. In total, 6240 trials were retained for analysis (20 participants x 2 conditions x 2 blocks x (9+4) targets x 3 fingers x 2 repetitions). In the swiping study, each block consisted of a single set composed of two repetitions of each of the three fingers completing a stroke from one on-screen target to another in each of the four cardinal directions. The required stroke distance was always 2cm. In total, 1296 trials were analyzed (9 participants x 2 conditions x 3 blocks x 3 fingers x 2 repetitions x 4 directions). Stroke direction was not treated as an independent variable.

For each trial we recorded: the preparation time, the span between the start of each trial and the first touch to the screen; the touch time, or period in which the finger was in contact with the screen; the error rate in terms of successful completion of the requested interface operation (e.g. selecting a button) and; the stream of 8x8 raw sensor data. In order to acquire reliable data, all touches were required to generate at least three packets of sensor data – given the

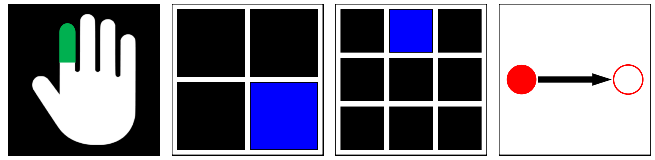

Figure 3. Study instructions. Left: the icon used to index what finger should touch the screen (index in the example). Center: the 2x2 and 3x3 button grids, each with one target highlighted in blue. Right: an example swipe instruction to drag the red target to the white one. All were shown on the 33mm touch screen area used in the studies.

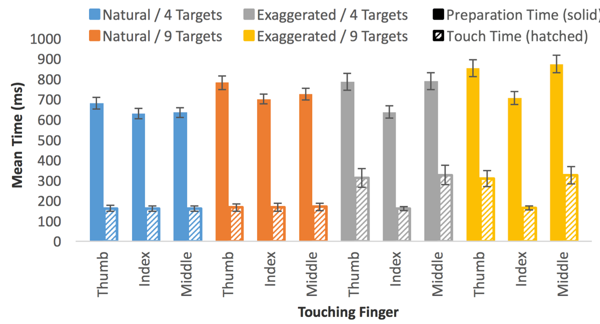

Figure 4. Mean Preparation and touch times from the tap study by finger and condition. Bars show standard error.

system's 33Hz update rate, this meant ~90ms of touch time. In trials when the user touched the screen for less than that time, they were required to repeat the trial, and no data or targeting error was recorded. The NASA TLX was used to capture workload after the natural and exaggerated conditions were completed in both studies.

Performance Results

Initial analysis of the data focused on the fundamentals of performance: time, errors and workload. Specifically, we report on time and error data per finger in the natural condition in order to contrast this smartwatch data with that derived from related studies on larger form factors such as tablets [4, 21]. We also compare data in the natural condition with that in the exaggerated condition to understand the impact of requiring the user to adopt specific finger poses. Finally, in the tap study, we also examine the differences between the small and larger targets. All analyses, except where otherwise mentioned, were factorial RM ANOVA incorporating Greenhouse-Geisser corrections to adjust for sphericity violations and followed-up, if required, by post-hoc t-tests incorporating Bonferroni corrections. We also report effect size for ANOVA results as partial-eta squared (η_p^2). For brevity, only significant results at $p<0.05$ are reported.

Figure 4 shows the preparation and touch time data from the tap study. Preparation time showed two significant interactions: number of targets by finger ($F(2, 38) = 4.094$, $p<0.05$, $\eta_p^2=0.177$) and condition by finger ($F(2, 38) = 40.44$, $p<0.001$, $\eta_p^2=0.68$). All main effects were also significant: condition ($F(1, 19) = 20.067$, $p<0.001$,

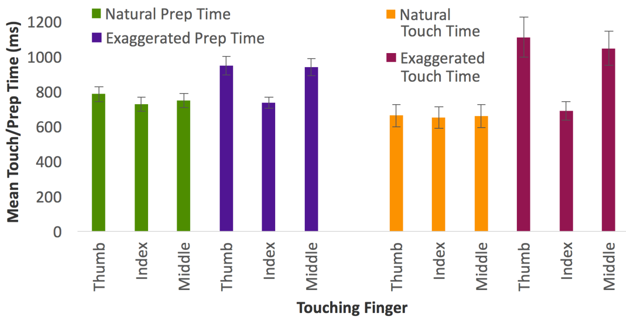

Figure 5. Mean Preparation and touch times from the swipe study by finger and condition. Bars show std. error.

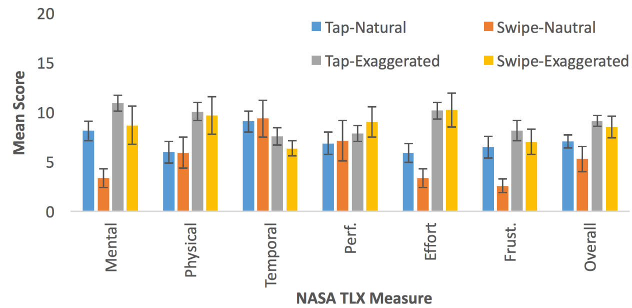

Figure 7. Mean TLX workload scores from both tap and swipe studies by condition. Bars show standard error.

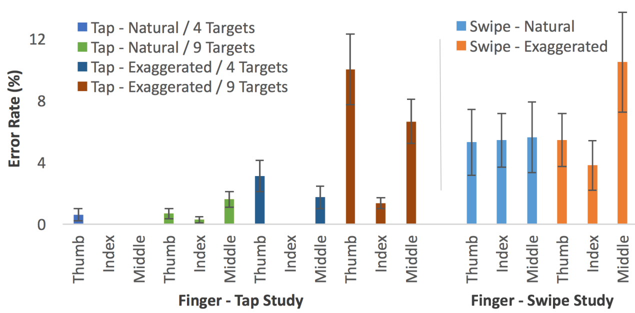

Figure 6. Error rates from both tap and swipe studies by finger and condition. Bars show standard error.

η_p^2=0.521), number of targets (F (1, 19) = 111.427, p<0.001, η_p^2=0.854) and finger (F (1.46, 27.8) = 58.5461, p<0.001, η_p^2=0.775). Interpreting these results in terms of the three strongest effects, we can say that selections of the smaller targets required more preparation time and that this effect was stronger in the exaggerated condition and specifically with the thumb and middle fingers. The touch time data showed fewer differences, but a similar story. Only the condition by finger interaction (F (1.049, 19.93) = 15.899, p=0.001, η_p^2=0.456) and main effects of condition (F (1, 19) = 10.404, p<0.01, η_p^2=0.354) and finger (F (1.055, 20.042) = 15.912, p=0.001, η_p^2=0.456) attained significance. This suggests that the interaction is the key effect in this case and the difference can be simply explained by the increased touch time in the thumb and middle finger trials in the exaggerated condition.

Figure 5 shows preparation and touch time for the swipe study. All preparation time and touch time comparisons were significant. For preparation time the figures are: interaction (F (2, 16) = 21.706, p=0.001, η_p^2=0.732) and main effects of condition (F (1, 8) = 23.186, p=0.001, η_p^2=0.743) and finger (F (2, 16) = 26.627, p=0.001, η_p^2=0.769). For touch time, these data are: interaction (F (1.27, 9.734) = 32.687, p=0.001, η_p^2=0.803) and main effects of condition (F (1, 8) = 21.151, p=0.002, η_p^2=0.726) and finger (F (1.147, 9.177) = 31.213, p=0.001, η_p^2=0.796). The interactions are again the dominant effects and these data reinforce the findings from the swipe study that the use of exaggerated poses for the thumb and middle fingers negatively impacted task completion times.

Error rates are shown for both studies are shown in Figure 6. Errors in the tap study did not feature normal distributions – the mode for all bar one combination of conditions was zero. As such, we analyzed these data with three separate Friedman tests, one for each variable. All three returned significant results: finger (χ2(2) = 12.5 p = 0.002), number of targets (χ2(2) = 20.0 p < 0.001) and condition (χ2(2) = 10.889 p = 0.001). Follow-up Wilcoxon tests indicated that the index finger resulted in significantly lower error rates than the thumb (Z = -3.057, p = 0.002) and middle finger (Z = -3.7, p < 0.000). Beyond confirming the additional challenge of smaller targets, these results also indicate that tap performance is optimal with the index finger and natural input condition. Although we were not able to examine interactions, the chart suggests that these effects are largely due to the spike in error rates with the thumb and middle finger when completing trials with the smaller targets. Error rates at other times remain relatively low. In contrast to these variations, error data in the swipe study were fairly flat. They were also somewhat higher, most likely due to the fact the compound dragging task was more challenging, and distributed more normally. As such, we analyzed them with a single three-way RM ANOVA. However, no comparisons in the swipe study reached significance at the p<0.05 level.

Finally, TLX data are shown in Figure 7. In the interests of brevity, we describe overall workload as a representative measure. This data was analyzed with matched pairs t-tests to contrast performance in the natural and exaggerated conditions in both studies. The results indicate that the natural conditions received lower ratings of workload than the exaggerated conditions (both p<0.001), mirroring the trend suggested in the chart.

Classifier Results

In order to build a finger classifier for the touch images, we first selected a single touch image from the temporal center of the data for each trial. This is because finger touch profiles at the start and end of a touch (the moment a finger touches or release the screen) may vary substantially from

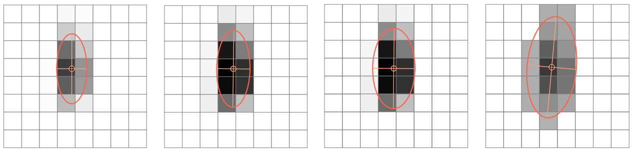

Figure 8. The four touch images used for classification. Left: raw-image, left-center: power3, right-center: thresholded-power3 and right: thresholded-log. Red ellipses are calculated from each figure's image moments.

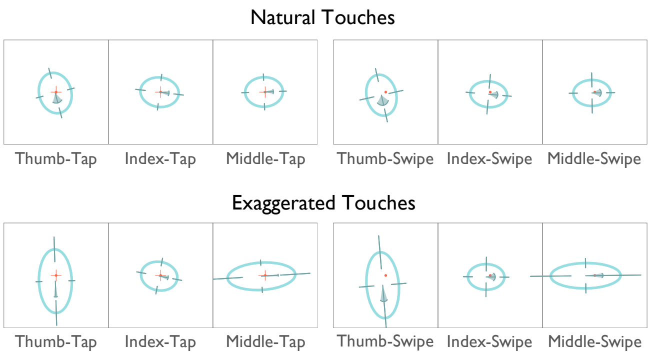

Figure 9. Mean ellipses from the raw touch image for each study and finger. Standard deviations for angle and major/minor axis length are shown via the arc and bars. The red dot marks center of the target and displacement from the center of the ellipse marks the mean center position; red bars show its standard deviation.

those during the middle portion of a touch [26], when the finger is fully in contact with the screen surface. We wanted to exclude these transient data points. We then generated ellipses for all these touches using both blob tracking [18] and image moments [29] approaches. Ellipses were defined as angle, major and minor length and eccentricity. In the tap condition, we also recorded the x and y center with respect to the current target, while in the stroke condition we just logged the raw center position. Visual inspection revealed the image moments led to ellipses that better matched the raw data, most likely due to the fact that the blob tracking approach thresholds the image to black and white rather than considering it as a greyscale image. Following prior authors [10, 29] we also applied several gamma corrections to the image to enhance the ellipses, although we found different parameters more effective. Specifically, we created three adjusted images: transformed to the power three; thresholded at 5% of the maximum reported data value then transformed to the power three and; thresholded and log transformed. Figure 8 depicts an example of the four touch images generated and Figure 9 shows the mean ellipses derived from all the raw images for both natural and exaggerated touches in both the tap and swipe studies.

We used this data to construct recognizers for the touching finger using Weka [6]. All recognizers were built using a ten-fold cross validation process and Random Tree or Random Forest decision trees. We selected these techniques as they are mature and relatively quick to execute (so suitable for small devices). In the following description figures and statistics are included for clarity, but we note that Table 1 summarizes all the content reported in terms of recognizers, datasets, attributes and results.

Visual inspection of the raw touch data indicates touches in the exaggerated condition are highly distinctive, while those in the natural condition show substantial overlap. Accordingly, we first constructed static recognizers based on all data from the exaggerated conditions in each study. For the exaggerated tap data, class-wise histograms showed the attributes of eccentricity and orientation had high discriminatory power. We used these attributes to construct a simple three level Random Tree and achieved a mean accuracy of 98%. However, applying the same approach to the swipe data led to a lower mean accuracy: 92.3%. In addition, class-wise performance varied considerably (Kappa: 0.88), with the middle finger at 96.3% accuracy and the index finger at 86%. This suggests that the dynamic touches in the swipe study are harder to classify than simple taps. In order to increase performance, we constructed a 10 tree Random Forest using the full description of the ellipses from the raw data set: position, size, angle and eccentricity. This attained a mean accuracy of 97.7% (Kappa: 0.96). We believe these figures are sufficiently high to reliably identify fingers if users are instructed to use exaggerated touches on a smartwatch.

Unsurprisingly, applying these relatively simplistic approaches to data from the natural conditions, where we expect both a greater diversity of touches and a less distinctive set of features, resulted in lower accuracies. To boost accuracy, we first created static models using all ellipse attributes from our four data sets, an approach similar to Xiao et al. [29], and increased the number of Random Forest trees to 100. For the tap data, this led to an overall accuracy of 68%. Class-wise performance was split (Kappa: 0.52) with the thumb at 88.1% and the index and middle fingers showing lower performance (53.8% and 62.1%, respectively). Following Wang et al.'s [26] observation that touch profiles on screen are time varying, we also examined performance with this recognizer on the subset of the tap data extracted from longer trials – those that recorded at least 150ms, or five packets of data. In total this was 1568 trials (50.2% of the original set), spread evenly over the three finger classes. This led to a modest improvement in the mean recognizer performance to 70.6%. Finally, we explored the impact of individual differences on performance by generating separate per-user models (using a 10-tree random forest on all data from the normal touch image) yielding a mean accuracy of 79.4%, again with best performance for the thumb (93.4%) and lower performance for the index (70.3%) and middle (74.4%) fingers. Applying these same approaches to the swipe data led to lower figures: the mean accuracy of the per-user models was 72.1%, maintaining the trend in which the thumb is more distinctive (83.8%) than the index (65.3%) and middle (67.2%) fingers.

Discussion

Mean performance in the tap study was relatively fast and accurate compared to prior work documenting performance on smartwatches [e.g. 13]: overall mean task time was 947ms and error rates were 2.1%. The more complex swipe actions, which took the form of a drag between two targets, took longer and yielded more errors: 1617ms and 6%, figures that are again consistent with prior work [4]. It is informative to compare this data with Goguey *et al.*'s [4] and Roy *et al.*'s [21] recent examinations of different finger input on tablets. Differences in study design and objectives make direct time comparisons challenging (Roy terminates time measurement on first screen contact, while Goguey's work examines the span of a pair of screen contacts), but both articles report performance variations for different fingers. Data on thumb, index and middle during tapping and stroking reveals temporal and error data follow the V-shape also observed in the current study: performance is optimal with the index finger. Error rates for tap in these articles are reported in the range of 1.9%-3.2%, figures consistent with all bar the smallest targets in the exaggerated input condition. These comparisons suggest that performance in the current study was typical and serves to confirm that findings on finger input reported in prior work on tablets also applies to the smartwatch form factor.

Beyond establishing this baseline, the main objective of capturing time/error data in the current studies was to explore the costs incurred by requiring users make specific finger poses during interaction. These costs are clear: in the tap study, preparation times edge upwards (from 704ms to 824ms) when users have to make touches in specific poses; touch times nearly double (from 167ms to 318ms) as they actually make these contacts. In the swipe study, time data show similar trends. Error data tells a somewhat more complex story, with data remaining relatively flat when tapping large targets and during swipes and spiking dramatically in selection tasks with smaller targets. This suggests that finger identification techniques based on touch contact profiles might be best applied to tasks involving coarse-grained targeting actions. Finally, TLX data confirm these variations caused participants to feel increased levels of workload in the exaggerated conditions.

Despite these costs, we note that performance remains within acceptable levels in exaggerated study conditions: mean task times of 1142ms (tap) and 2022 (swipe) and error rates of ~0-5%. Furthermore, variations in workload (and other data) need be treated with caution due to the fact that the natural condition always preceded the exaggerated condition. While this design prevented the exaggerated instructions from biasing natural behavior, it may mean that fatigue is artificially inflating differences. Evidence to support this idea comes from the fact that the lowest workload scores were recorded in the first condition administered in the more complex but shorter swipe study. We also note that workload levels remain generally low – from 3-10 out of 20 – across the whole study. This suggests participants never felt the tasks to be high demand. In sum, while the costs of using specific poses for finger identification are clear, their magnitude is relatively limited. While these costs may make the technique unsuitable for highly frequent interactions (like the repeated and prolonged tasks in the studies), we argue they likely remain acceptable for the more sporadic use scenarios that are more typical of genuine device operation in the real world.

Recognizer	*Attributes*	*Touch Image(s)*	*Condition / Data Set*	*Accuracy*				*Kappa*
				Mean	**Thumb**	**Index**	**Middle**	
Random Tree (Static, 3 deep)	Eccentricity, Orientation	Normal	Tap-Exaggerated	98.6%	98.5%	96.4%	99.1%	0.97
			Swipe-Exaggerated	92.3%	94.5%	86%	96.3%	0.88
			Tap-Natural	61.9%	82%	12.6%	91.1%	0.43
			Swipe-Natural	55.3%	80.6%	71%	14%	0.32
Random Forest (Static, 10 trees)	Eccentricity, Orientation, Major, Minor, CenterX, CenterY	Normal	Swipe-Exaggerated	97.7%	98.6%	99.1%	95.3%	0.97
			Tap-Natural	67%	85.2%	50.6%	65.3%	0.50
			Swipe-Natural	62.7%	80.1%	53.3%	54.7%	0.44
Random Forest (Static, 100 trees)	Eccentricity, Orientation, Major, Minor, CenterX, CenterY	Normal, Power3, Thresholded-Power3, Log	Tap-Natural	68%	88.1%	53.8%	62.1%	0.52
			Swipe-Natural	65.4%	85.6%	55.6%	54.7%	0.48
			Tap-Natural - Long	70.6%	90.7%	55.6%	66.3%	0.56
Random Forest (Per-user, 10 trees)	Eccentricity, Orientation, Major, Minor, CenterX, CenterY	Normal	Tap-Natural (mean result)	79.4%	93.4%	70.4%	74.7%	0.69
			Swipe-Natural (mean result)	72.1%	83.4%	65.3%	67.2%	0.58

Table 1. Results from the machine learning models constructed to analyze touch shape during tap and swipe.

The flipside of documenting these costs to performance is an exploration of the benefits of exaggerated finger poses for finger recognition. These are powerful. Recognizers in the exaggerated conditions are simple and high accuracy (~98%). This suggests that touches made under these constraints could be effectively deployed in real interfaces. Results for recognizers in the normal conditions are less clear cut. Although the 72.1% (swipe) to 79.4% (tap) mean accuracies achieved in models created for each user show promise, they are clearly insufficient for deployment in realistic interfaces. Performance with static models covering the whole participant population is worse yet. However, examining the class-wise error rates suggests the picture is more nuanced. Specifically, the thumb can be identified relatively reliably (up to 93.4% with the per-user models) while the index and middle finger remain challenging to distinguish (70.4%-74.7%). Furthermore, the confusion matrix (not pictured) from the per-user tapping models reveals that thumb and index finger taps are cross-classified only 2.8% (thumb as index) and 3.1% (index as thumb) of the time. This indicates that it may be possible to create reliable finger identification systems using our approach if only the thumb and index are considered. As such, we note that while it would be infeasible to use our system to directly enable interfaces such as Gupta and Balakrishnan's [5] index plus middle finger keyboard, it might be possible to adapt these interfaces to leverage the optimal performance of our recognizers via, for example, the use the thumb, or by requiring exaggerated touches. This observation highlights a critical point: the design of finger recognition interfaces needs be informed by the properties of the underlying recognition system.

The remainder of this paper explores this issue: given the user performance constraints captured and recognizer accuracies documented, what interface designs are useful, effective and feasible? We investigate this issue in two ways. Firstly, inspired by the DualKey system [5], we implement and study two finger-identification powered keyboards. The goal of this work is to understand real world performance with the static recognizers and input modes documented and proposed in this paper. Rather than just relying on the data from the studies, we use this complex task to push the boundaries of the system and observe how recognizer and user performance instantiate and interact in a more realistic task incorporating, for example, immediate graphical feedback relating to the outcomes of a user's actions. We close by providing a more general discussion that consolidates all the work in the paper into practical recommendations and design examples.

DI-TYPE AND TRI-TYPE KEYBOARDS

We created two smartwatch virtual keyboards that use static natural (100 tree random forest) and static exaggerated (random tree) tapping models generated from the data captured in the first tapping study. For natural taps, we created Di-Type, a dual finger design, with two letters marked on every key. Tapping with the thumb recorded the leftmost letter and tapping with the index or middle finger recorded the rightmost letter. This reflects the fact that the natural model recorded higher accuracy for the thumb and lower scores for the index and middle fingers. For the exaggerated touches, we placed three letters on each key to create Tri-Type. The leftmost key was activated by the thumb, the center one by the index and the right one by the middle finger. This keyboard requires participants to mimic the touch poses used in the exaggerated conditions of the studies. The keyboards were both ordered alphabetically to facilitate novice users in the task of locating letters. Where possible we also used common key arrangements, such as the 3x3 arrangement of letter triples in Tri-Type (see Figure 10). Both keyboards were 33 x 33mm in size. They featured a 2.1mm border, an inter-key spacing of 0.6 mm and a text display bar at the top with a height of 4mm. The buttons divided up the remaining space equally: 6.75mm x 5.75mm and 6.75mm x 7.85mm, respectively, for Di-Type and Tri-Type. They are both shown in Figure 10.

We performed a limited evaluation of these systems with 11 participants (mean age 21, six female). All participants entered 30 randomly selected sentences from Mackenzie *et al*'s [14] phrase set using both keyboards. As with the earlier studies participants always used the natural Di-Type system first. The first 15 phrases entered with both keyboards were considered practice and not analyzed. During their initial use of the keyboards, participants were encouraged to explore the keyboard, the novel finger-identification input scenario and the finger recognition process freely for up to 30 minutes. In the natural condition, they were not given formal instructions on how to touch the keyboard, but an experimenter did demonstrate how to use

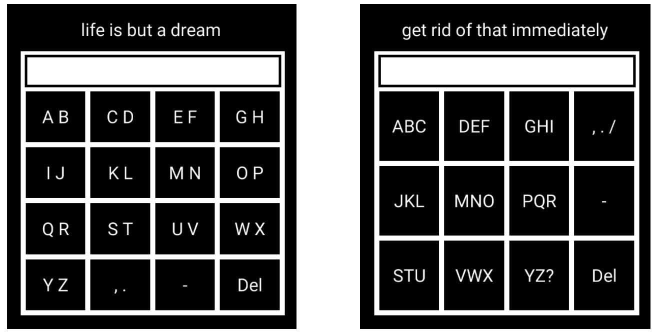

Figure 10. Di-Type (left) and Tri-Type (right) keyboards. Thumb taps selects the leftmost key. In Di-Type, the index or middle finger select the right key. In Tri-Type, index selects the center key and middle the right key.

the keyboard if requested. In the exaggerated condition, the finger poses were demonstrated to participants. In total the experiment took approximately 90 minutes per participant and each was compensated with ~15 USD in local currency.

The goals of this study were more focused on validating the recognition performance than on text entry performance. As such we logged raw Words Per Minute (WPM) to support a basic comparison with prior work [5] and asked participants not to correct any errors. The primary measures were then calculated from the text streams. We classified each character as either correct, or as a *wrong-key* error (meaning the wrong keyboard key had been selected) or a *wrong-finger* error (meaning that the wrong finger had been used or recognized). If participants entered additional or insufficient characters in a string, this was treated as an error and they were required to enter another string in order to complete the study. They could also tap the top of the keyboard to cancel a trial at any time.

Results and discussion

Over the course of the study a total of 8164 characters were entered and retained for analysis. Participants achieved a mean of 8.08 (SD 0.78) raw WPM with Di-Type and 7.53 (SD 0.87) raw WPM with Tri-Type, figures that a paired t-test revealed were not significantly different from one another. These figures are somewhat slower than those recorded in the initial sessions of Gupta and Balakrishnan's [5] DualKey – they report mean WPMs of around 10.8. There are many possible explanations for this. One is that the finger recognition system used in the studies took longer for users to operate because of, for example, its reliance on the thumb, or its use of three fingers or specific poses. We also note that while our participants were engaged in study at an institution whose language of instruction is English, none were native readers of Latin characters. A dedicated comparison study with an implementation of both systems would be required to understand the cause of these differences. Instead, we note that the WPM figures indicate participants were able to type using both systems at a reasonable speed on a tiny screen.

More interesting are the error results. First, we recorded a mean of 1.45 (SD 1:87, median 1) sentences with an incorrect number of characters. Participants also cancelled entry processes when they observed extra characters – on average 4.13 times, a distribution skewed by one participant who frequently performed this behavior (SD: 8.2, median: 1). We did not analyze these trials further, but their presence does indicate that participants did at times cancel tasks on noticing errors involving entry of extra characters, potentially skewing the data towards more successful trials. Given the relatively infrequent occurrence of this behavior, we do not believe it exerted a strong effect on the study. Wrong-key errors were low thorough the study, running at means of 1.7% (SD: 1.2%) with Di-Type and 1% (SD: 0.9%) with Tri-Type. A matched t-test revealed these figures were not significantly different. This indicates that participants were able to select the small keys used in the keyboards with a very high degree of accuracy and regardless of the use of the exaggerated touch pose. Wrong-finger errors were more commonplace. Means per participant were 8.3% (SD 4.4%) with Di-Type and 10% (SD 3.8%) with Tri-Type. We note these figures include genuine input mistakes – situations when a user actually tapped with the wrong finger. A paired t-test revealed no difference in the wrong-finger rate between the keyboards.

Figure 11 shows the overall mean wrong-finger rate (per key) for each keyboard and finger. There are substantial variations among the fingers, with the index finger being recognized most accurately. Looking at the data in detail, we observed a disproportionate number of thumb wrong-finger errors in the bottom row of keys. In the single Di-Type letter key on the bottom row, thumb input logged a 42% wrong-finger rate. In Tri-Type, this ran at a mean of 24% wrong-finger errors for the bottom three keys. This suggests it was highly challenging to correctly identify the thumb at the bottom of the screen, an effect that was likely not observed in the main studies due to their larger targets. If errors from the bottom row are removed overall wrong-finger rates for the thumb drop from 10% to 7.3% for Di-Type and, more substantially, from 9.7% to 2.3% for Tri-Type. This problem impacted participants: in post-study comments, they indicated a preference for Di-Type due to a combination of the fewer fingers required to operate it and problems acquiring bottom row targets with Tri-Type.

In sum, this study shows our approach to finger identification has considerable promise. Static finger identification models generated from the tapping data of one set of participants enabled a second set to reasonably successfully and rapidly enter text, a challenging input task. Although mean accuracy for each keyboard was in the range of 90%-92% (see Figure 11) and clearly lagged behind systems that use dedicated hardware [5], we note that these figures incorporate actual user mistakes and also expect that redesigning the keyboard to avoid trouble spots such as the bottom of the screen and integrating customized per-user recognizers can improve performance in the future.

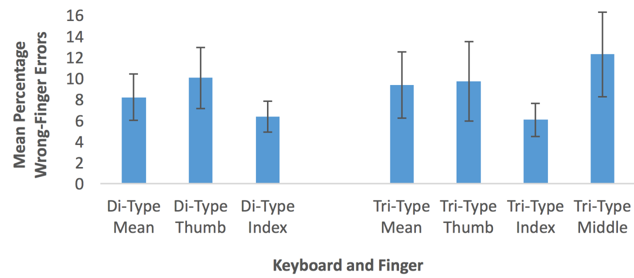

Figure 11. Mean percentage wrong-finger errors (by key) in Di-Type & Tri-Type keyboards. Includes mean and per finger data for both keyboards. Bars show Std. Dev.

RECOMMENDATIONS AND DESIGNS

Moving beyond this validation, the user and recognizer performance data captured in the studies are rich enough to support a range of practical recommendations about how finger identification using capacitive touch profiles could be best applied to designing interfaces on smartwatches. We break these down into key themes.

Fingers

Prior authors have documented how performance with different fingers varies in touch tasks [4, 21]. The current work confirms this is also true on smartwatches. However, recognizer performance also impacts design choices. The work in this paper clearly indicates that taps with the thumb are distinct from other taps – relatively high recognition accuracy could be achieved with no prior instructions and no changes in task performance and workload. Therefore, any system requiring only two fingers should first consider a design that discriminates between the thumb and other digits as the most practical and comfortable approach. Furthermore, results (and stated preferences) from our Di-Type and Tri-Type prototypes seem to suggest that less is more and that systems should use as few fingers as possible in order to achieve their objectives. Operating a system with two fingers is easier for users to deal with than three.

Targets

In the main studies in this paper, performance with thumb and middle finger input decreased with smaller targets; index finger input was unaffected. This trend was particularly prominent with the exaggerated poses. This suggests that finger identification technology can be most effectively applied to relatively large targets. On smartwatches, with their limited screen space, this may serve to restrict effective systems to specific types of content such as application icons, or continuously available actions such as a back function. While this effect was not evident in the final typing study, we did observe that recognition performance dropped substantially during thumb touches on the bottom of the screen – another risk for small targets is that proximity to screen edges may mean that full touch contact areas cannot be accurately captured. If small targets are used (and the typing study suggests that may be viable), then they should be situated away from the bottom and right edges of a watch screen.

Actions

Slow and simple touches can be more accurately recognized. Specifically, in the natural tapping study, the fingers making touches over 150ms were classified 2.8% more reliably the full set of touches. Due to the length of touches captured in that study (mean of 150ms for the index finger) it was not possible to explore whether longer touches achieved greater gains. However, we also note that more complex actions, even if they are prolonged (such as the movements in the swipe study), will likely result in greater variability in touch contact area and lower classification accuracy. To be reliable, finger identification systems should therefore rely on techniques such as dwell thresholds before triggering classification processes. This kind of technique may also serve to lower false positive rates – touches under a certain duration are all treated as the default regardless of the touch contact area. Dwell thresholds could also be used to combine finger identification with more complex input techniques like swipe – classification could take place during the dwell and a subsequent movement could then further specify input.

Building on these discussions we present two interactions designs that adapt themes for interaction design with finger identification systems presented in prior work. The first of these is *tricons*, an idea that relates to the multiple finger icons discussed by, for example Roy *et al.* [21]. Tricons enhance applications icons by providing multiple points of entry. For example, a fitness application could be opened as normal using a regular tap, have an exercise routine start with a middle finger tap and open settings with a thumb tap. Equally, a clock could access alarm, timer or main functions depending on the tapping finger. This kind of icon matches our design recommendations as they are relatively large, accessed sporadically and usually situated away from the extreme edges of the screen. Figure 12 (left) shows two possible tricon designs. The second design is a context menu, similar to those proposed by Harrison *et al.* [7]. We envisaged this design for a music player and operating as follows. A user calls up the menu with an easily recognizable thumb tap that is held against the screen. After a short dwell, a context menu appears around the thumb and subsequent horizontal swipes navigate between tracks while vertical movements adjust volume (Figure 12, right). All normal controls of the smartwatch that respond to regular touches are unaffected. Rather than as fully novel contributions, we present these designs as customized versions of existing concepts that fit the capabilities of the functional finger recognition system proposed in this paper.

DISCUSSION AND CONCLUSIONS

This paper contributes the idea of recognizing the finger touching a smartwatch from the profile it generates on the device's capacitive touch screen. It also contributes data that explores the tradeoffs between finger recognition accuracy and user input performance in natural and exaggeratedly posed touches and validates its approach in a challenging text entry task. Taken together this work is a first characterization of the feasibility of finger identification using standard smartwatch touch screens and a comprehensive investigation of the practical limitations of the technique – its strengths, weaknesses and how these impact what can be built with it.

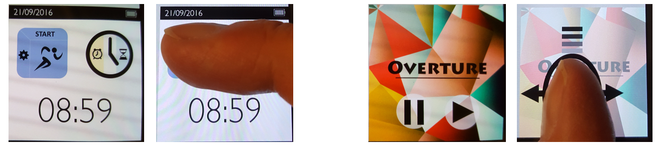

Figure 12. Example finger identification enabled interfaces. Left shows two *tricons*, icons that respond differently to each finger. Center left shows a middle finger touch that would start an exercise routine on an activity tracker. Center-right depicts a typical music player interface. Right shows how a touch with the thumb can bring up a menu to switch between tracks (swipe left/right) or adjust volume (swipe up down).

There are a number of interesting future avenues for research. This paper focused on the recognition of fingers from single frames of touch data. Exploring features than span entire screen contacts is an obvious next step. A prerequisite for achieving this is likely a faster sensor – the 33Hz sensor data used in this work resulted in rapid touches leaving few records. More empirical work is also required in terms of data capture – the lab studies in this paper suffer from typical issues of ecological validity. While we did focus on a common pose (seated at a desk), wearables are clearly used much more diversely. Contexts such as use on public transport or during discrete operation under a desk or on a lap are valid and worth studying. To do so will require porting the system to a genuine smartwatch (as in [29]) and this is a clear next step for this project. We also note that the work in this paper deals with sensing finger touch profile and angle – it may be possible for users to operate the system simply by angling their fingers appropriately to form different shapes [1, 18]. Exploring how these closely related input modalities could complement one another would be another interesting next step for this work.

In conclusion, finger identification is a simple, effective input technique that can yield many benefits on wearables. This paper provides a first examination of how it might be enabled using standard capacitive touch screen technology. We believe the ideas, techniques and recommendations we present can guide designers and developers as they introduce finger identification into real devices.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2014R1A1A1002223) and the ICT R&D program of MSIP/IITP. [R0190-15-2054, Development of Personal Identification Technology based on Biomedical Signals to Avoid Identity Theft]. We would also like to thank our anonymous reviewers and ACs, whose detailed and insightful comments helped us to improve this paper.

REFERENCES

1. Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The fat thumb: using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (MobileHCI '12), 39-48. http://dx.doi.org/10.1145/2371574.2371582
2. Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. 2012. Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (ITS '12), 173-182. http://dx.doi.org/10.1145/2396636.2396663
3. Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2013. Arpège: learning multitouch chord gestures vocabularies. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces* (ITS '13), 209-218. http://dx.doi.org/10.1145/2512349.2512795
4. Alix Goguey, Mathieu Nancel, Géry Casiez, and Daniel Vogel. 2016. The Performance and Preference of Different Fingers and Chords for Pointing, Dragging, and Object Transformation. In*Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 4250-4261. http://dx.doi.org/10.1145/2858036.2858194
5. Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 59-70. http://dx.doi.org/10.1145/2858036.2858052
6. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* 11, 1 (November 2009), 10-18. http://dx.doi.org/10.1145/1656274.1656278
7. Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 627-636. http://dx.doi.org/10.1145/2047196.2047279
8. Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 581-590. http://dx.doi.org/10.1145/1753326.1753413
9. Christian Holz and Patrick Baudisch. 2013. Fiberio: a touchscreen that senses fingerprints. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13), 41-50. http://dx.doi.org/10.1145/2501988.2502021
10. Christian Holz, Senaka Buthpitiya, and Marius Knaust. 2015. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. In*Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 3011-3014. http://dx.doi.org/10.1145/2702123.2702518
11. Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. 2016. Faster Command Selection on Touchscreen Watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 4663-4674. http://dx.doi.org/10.1145/2858036.2858166
12. Gierad Laput, Robert Xiao, Xiang 'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 389-394. http://dx.doi.org/10.1145/2642918.2647356
13. Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 669-678. http://dx.doi.org/10.1145/2702123.2702388
14. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '03), 754-755. http://dx.doi.org/10.1145/765891.765971
15. Nicolai Marquardt, Johannes Kiemer, and Saul Greenberg. 2010. What caused that touch?: expressive interaction with a surface through fiduciary-tagged gloves. In *ACM International Conference on Interactive Tabletops and Surfaces* (ITS '10), 139-142. http://dx.doi.org/10.1145/1936652.1936680
16. 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. http://dx.doi.org/10.1145/2556288.2557138
17. Ian Oakley, DoYoung Lee, MD. Rasel Islam, and Augusto Esteves. 2015. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1237-1246. http://dx.doi.org/10.1145/2702123.2702226
18. Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD. Rasel Islam. 2016. The Flat Finger: Exploring Area Touches on Smartwatches. In *Proceedings of the*

2016 CHI Conference on Human Factors in Computing Systems (CHI '16), 4238-4249. http://dx.doi.org/10.1145/2858036.2858179

19. Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). 1451-1460. http://doi.acm.org/10.1145/2470654.2466192
20. Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 2575-2584. http://doi.acm.org/10.1145/1978942.1979318
21. Quentin Roy, Yves Guiard, Gilles Bailly, Éric Lecolinet, and Olivier Rioul. "Glass+ skin: An empirical evaluation of the added value of finger identification to basic single-touch interaction on touch screens." In *Human-Computer Interaction*, pp. 55-71. Springer International Publishing, 2015. http://dx.doi.org/10.1007/978-3-319-22723-8_5
22. T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making muscle-computer interfaces more practical. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 851-854. http://dx.doi.org/10.1145/1753326.1753451
23. Craig Stewart, Michael Rohs, Sven Kratz, and Georg Essl. 2010. Characteristics of pressure-based input for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 801-810. http://dx.doi.org/10.1145/1753326.1753444
24. Atsushi Sugiura and Yoshiyuki Koseki. 1998. A user interface using fingerprint recognition: holding commands and data objects on fingers. In *Proceedings of the 11th annual ACM symposium on User interface software and technology* (UIST '98). ACM, New York, NY, USA, 71-79. http://dx.doi.org/10.1145/288392.288575
25. Jingtao Wang and John Canny. 2004. FingerSense: augmenting expressiveness to physical pushing button by fingertip identification. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '04), 1267-1270. http://dx.doi.org/10.1145/985921.986040
26. Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. 2009. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 23-32. http://doi.acm.org/10.1145/1622176.1622182
27. Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 2991-3000. http://dx.doi.org/10.1145/2702123.2702391
28. Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 3847-3851. http://dx.doi.org/10.1145/2858036.2858466
29. Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (ITS '15), 47-50. http://dx.doi.org/10.1145/2817721.2817737
30. Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, and Aaron Quigley. 2016. WatchMI: pressure touch, twist and pan gesture input on unmodified smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16), 394-399. http://dx.doi.org/10.1145/2935334.2935375
31. Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15), 167-173. http://dx.doi.org/10.1145/2807442.2807480
32. Cheng Zhang, Junrui Yang, Caleb Southern, Thad E. Starner, and Gregory D. Abowd. 2016. WatchOut: extending interactions on a smartwatch with inertial sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers* (ISWC '16), 136-143. http://dx.doi.org/10.1145/2971763.2971775