Beats: Tapping Gestures for Smart Watches

Ian Oakley, DoYoung Lee and MD. Rasel Islam
Department of Human and Systems Engineering
Ulsan National Institute of Science and Technology, Ulsan, Korea

Augusto Esteves
Madeira-ITI, University of Madeira, Funchal, Portugal

ian.r.oakley@gmail.com, ehdud611@gmail.com, islam@unist.ac.kr, augustoeae@gmail.com

ABSTRACT
Interacting with smartwatches poses new challenges. Although capable of displaying complex content, their extremely small screens poorly match many of the touchscreen interaction techniques dominant on larger mobile devices. Addressing this problem, this paper presents *beating gestures*, a novel form of input based on pairs of simultaneous or rapidly sequential and overlapping screen taps made by the index and middle finger of one hand. Distinguished simply by their temporal sequence and relative left/right position these gestures are designed explicitly for the very small screens (approx. 40mm square) of smartwatches and to operate without interfering with regular single touch input. This paper presents the design of beating gestures and a rigorous empirical study that characterizes how users perform them – in a mean of 355ms and with an error rate of 5.5%. We also derive thresholds for reliably distinguishing between simultaneous (under 30ms) and sequential (under 400ms) pairs of screen touches or releases. We then present five interface designs and evaluate them in a qualitative study in which users report valuing the speed and ready availability of beating gestures.

Author Keywords
Wearable; Smartwatch; Multitouch; Tapping

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Wearable technologies are entering the mainstream. Smartwatches, with a familiar and convenient wrist-mounted form-factor, are one of the main current device categories in this space. They promise to generate substantial value by enabling novel applications and services, specifically in areas such as bio-monitoring, health and the quantified self. Smartwatches can also act as helpers and adjuncts for larger mobile devices, providing notifications at a glance, the ability to remotely and conveniently issue basic commands (such as answering a call or silencing an alarm) and potentially more advanced functionality such as presenting app specific contextual menus or callouts that highlight key information [4]. Ultimately, they promise timely access to information and the ability to issue relevant responses rapidly and with minimal disruption to ongoing tasks.

However, despite this potential, their small size makes effective interaction with smartwatches a challenging task. Indeed, the basic device format – the watch – is historically a purely display device and not an interactive system. Current smartwatch designs reflect these origins: sharp, vivid high-resolution screens are paired with very limited and restricted interaction spaces. Recognizing this mismatch as both a problem and an opportunity, recent research prototypes and products have focused on introducing additional input capabilities to smartwatches. Example approaches include adding physical controllers such as dials to the watch form factor (e.g. the Apple Watch, www.apple.com/watch/), using the movements of watch itself as a physical controller [19] and extending input sensing to other surfaces of the watch, such as its edge [13] or strap [15]. While these techniques tackle the core screen-size problem directly and clearly have value, they also have limitations. Specifically, we note their novel input systems may prove difficult to integrate with the currently dominant touch screen paradigm [19] and it can be practically challenging to fit additional sensing hardware in the small, constrained spaces of the watch form factor.

Given these difficulties, we argue there is value in exploring novel interaction techniques that operate with the existing input surfaces of smartwatches – the touch screens. However, this will clearly require new approaches as the extremely small size of these modules practically precludes many of the techniques that have proven successful on larger systems, such as multi-finger pinch gestures [10]. Equally, the effects of well known foundational problems with touch screen interaction, such as the fact that finger touches obscure screen contents (the fat finger problem [17]), are likely to be exacerbated on smartwatches.

As one possible way of addressing these challenges, this paper proposes a new type of multi-finger input that is specifically designed for the very small touch screens of smartwatches. It is based on what we term *beating gestures*, pairs of simultaneous or rapidly sequential touches (and optionally one or more releases) made by the index and

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea
Copyright 2015 ACM 978-1-4503-3145-6/15/04...$15.00
http://dx.doi.org/10.1145/2702123.2702226

middle finger of one hand. Essentially, instead of tapping a single finger to a screen, a beating gesture involves adjacent screen contact (and optionally release) with two fingers and in three closely controlled intervals: either simultaneously or with one event immediately preceding the other as part of a single coordinated movement. The timings and crude relative position (e.g. left or right) of this input serves to characterize each movement and the fast pace of the paired touches distinguishes it from regular single finger input [9] or noise.

The motivations for exploring this specific design space include the fact that current smartwatch systems are large enough to register two adjacent touches and the simple sequences of movements required for beating gestures may be executable rapidly and eyes-free [3], avoiding fat-finger problems. Furthermore, we suggest the technique is sufficiently expressive to control a meaningful range of functions. Finally, as it is based on standard touch-screen input, but designed to not interfere with normal single finger use, we suggest it will be easy to integrate with existing on-screen interface styles such as single finger taps, swipes, dwell and gestures. Indeed, effective integration with established input techniques was a core design goal in our work.

The remainder of this paper sets out to fully explore the potential of beating gestures. The contributions include the core idea underlying beating gestures, a thorough quantitative characterization of how beating gestures are performed, a series of prototypes showing how they can be deployed to create realistic interfaces on a smartwatch and a qualitative study of how users react to these prototypes. Taken together this work represents a well-rounded description of beating gestures that can directly inform future research and provide both inspiration and practical guidance for interface designers creating novel touch interactions for smartwatches or other small screen devices.

RELATED WORK

Smartwatches are attracting increasing research interest in HCI as the combination of their wearable form factor, growing computational power and restricted input spaces make them a ripe target for the development of new forms of interaction. One trend has been to augment the devices with novel sensing systems on alternative surfaces. Perrault *et al.* [15], for example, created a watchstrap augmented with two touch sensors that enables users to make taps and strokes all around their wrists, while Oakley and Lee [13] propose co-opting the lateral edge of a watch-like device as an input surface by covering it with an array of touch sensors. There is also a wide range of work that proposes different mechanisms and approaches for Around-Device Interaction [11]. A prominent example in this space is Harrison and Hudson's elegant use of an in-device magnetometer in conjunction with an external permanent magnet mounted on a finger to support general pointing and interface tasks [6].

However, we argue that the largest and most accessible input surface on a smartwatch remains its screen. There has been relatively little work specifically exploring touch screen interaction design on smartwatches. Early exceptions include Blasko and Fiener's [2] and Ashbrook *et al.'s* [1] studies of how a beveled edge could be used to enhance target selection performance on watch displays. More recent work has explored how sensing finger orientation with an accelerometer could create a richer input space [4] and many techniques implemented on larger devices, such as identifying finger contact point [7] or shear [8] and pressure input [14] remain relatively unexplored on watches specifically but seem likely to offer considerable benefits.

Of particular relevance for this paper is prior work that has investigated systems based on rapid sequential taps. For example, Ghomi *et al.* [5] study how users can perform rhythmic Morse code like tapping patterns for tasks such as issuing commands or switching modes. They ultimately contribute a vocabulary of 14 tapping sequences that achieve a high level of usability. Serrano *et al.* [16] extend these ideas to discuss how a series of temporally separated taps that commences on the bevel of a tablet and moves over its screen can create a unique pattern of accelerometer disturbances and screen events that can be easily disambiguated from noise or other input and be used to, for example, summon menus and issue commands. Heo *et al.* [9] describe a closely related system based on input of spatially separated and rapidly sequential on-screen touches on a phone or tablet that can also be used for controlling menus or a typing-based form of gestural input. In designing their system, Heo *et al.* point out that a rapid sequence of touches is an unusual form of input and show that it is possible to temporally disambiguate taps in their system from regular touchscreen use.

The work in this paper is inspired by these prior studies and seeks to leverage the benefits of the tapping style of interaction (in terms of its support for rapid execution of commands and the fact it does not interfere with regular touch input) and apply this to the space-restricted smartwatch form factor. In order to do so this paper adapts these ideas into beating gestures composed of a rapid pair of simultaneous or overlapping taps. It explores how users produce these gestures and how they can be deployed to create useful interfaces on a smartwatch.

USER PERFORMANCE STUDY

To investigate the potential of beating gestures, we first conducted a study to capture, analyze and understand how users perform them. The primary goal of this study was descriptive - we specifically wanted to explore the full input space of beating gestures and establish parameters such as appropriate timing thresholds to distinguish between the different touch and release beating sequences. We also sought to determine whether the gestures would interfere with existing input paradigms and measure the reliability and consistency with which users could issue the gestures.

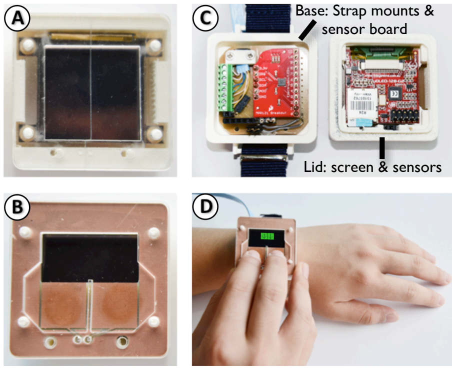

Figure 1. Watch prototype created for study. Plan views of watch screen with transparent ITO sensors (A) and half-screen PCB sensors (B). Right images show internal structure (C) and PCB prototype mounted on user's wrist with fingers touching the sensors and screen showing study UI (D).

Beats Hardware Prototype

Precise timing measurements, multi-touch input and the use of a wrist-based device were important aspects of this project. As multi-touch is not available on current watch format products, we constructed a prototype in the form factor of a smartwatch: a 3D printed 49mm by 49mm square by 25mm deep cuboid (Figure 1) with external mounting points for a standard watchband. These were sufficiently robust that the prototype could be securely attached to a user's wrist. The unit contained a 1.5inch (3.81mm) OLED screen (a 4DSystems μOLED-128-G2-GFX) and a Freescale Semiconductor MPR121 capacitive sensing microcontroller (specifically a Sparkfun MPR121 breakout board). The MPR121 was configured to report touches every four milliseconds, its highest possible update rate. Touches were captured on a pair of electrodes positioned over the surface of the screen. Two electrodes variants were developed – one full screen version based on transparent Itanium Tin Oxide (ITO) coated plastic sheets and a half-screen version using an opaque milled PCB board. While the ITO sheet offered the advantage of full visual availability of the display surface, it proved challenging to reliably calibrate its sensitivity to touches and its connections to the MPR121 board were frail and prone to failure. As such, we conducted the experimental work using the more robust and reliable PCB version of the system. A laser cut acrylic sheet was placed on top of the electrodes with holes that served to guide touches to their surfaces – without this sheet, small placement errors in single touches could result in contact with both electrodes and, thus, invalid input. Figure 1 illustrates these details.

The sensor and screen were connected via a single cable to an Arduino Mega by, respectively, I2C and RS232 communication links. The Arduino coordinated data logging and the low-level display of screen contents. It was connected by a second RS232 link to a PC and communicated sporadically (and not during touch time measurement) with a Java application that stored data and sent information about which trial contents to present.

Participants

Eighteen users completed the study (nine female, all right handed). They were undergraduate or graduate students enrolled in full time studies at UNIST in South Korea. Their mean age was 22 and they rated their experience with computers, touchscreens and smartphones to be high (4.7, 4.7 and 4.9 on a 1 to 5 scale) and their experience with smartwatches and other wearables devices to be low (1 and 1.3). Although the participant group was tech-savvy, in most cases this study was their first experience with a wearable computer. Those few with prior experience indicated they had used devices such as fitness trackers. Participants completed the study in 30-45 minutes and were compensated with approximately 10 USD in local currency.

Experimental Description and Design

The experiment sought to capture user performance in a broad set of beating gestures in order to establish both how reliably they can be produced by users and their temporal characteristics – the typical durations between each screen touch or release. As such we designed a complete set of beating gestures by systematically adjusting two variables: the sequences of finger *placement* and finger *release*. There are three possible placement sequences: the left finger followed by the right finger (*LR*), the right followed by the left (*RL*) and both together (*Dual-Tap*). There are six possible release sequences: holding both fingers on screen (*Hold*), releasing only the left finger (*L-Release*), only the right finger (*R-Release*), the left followed by the right (*LR-release*), the right followed by the left (*RL-release)* and both together (*Dual-Release*). Taken together these two variables lead to a set of 18 possible beating gestures.

The study was designed in four trial blocks. The first three each presented trials featuring a single finger-placement type (LR, RL or Dual-Tap). In each of these blocks, participants completed three trial-sets involving five randomly presented repetitions of each of the six possible finger-release sequences for a total of 90 trials. The first trial-set (e.g. the first 30 trials) were considered practice and not retained for analysis. This structure ensured that we captured participants' performance after they had gained experience with the study process and instructions and practiced each gesture for a short but sustained period of time, a manipulation intended to ensure that the results are more representative of experienced (rather than purely novice) performance. To mitigate practice effects with this structure, presentation of the finger-placement variable was fully balanced in a repeated measures design: three participants completed each of the six possible condition orders. In total, this stage of the study generated 3240 trials that were retained for analysis: 18 participants by 3 blocks by 2 trials-sets by 5 repetitions by 6 trials.

However, although we saw value in emphasizing non-novice performance, we were concerned that this study structure might lead to naturally unachievable mechanistic production of the beating gestures – simple repetition of the

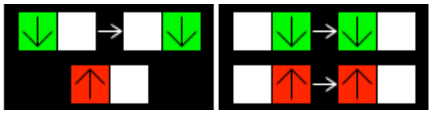

Figure 2. Two study instructions. Top row (green) indicates finger placements and bottom row (red) shows finger releases. Movement order is shown by horizontal position. The instructions are LR to L-Release (left) and RL to Release-RL (right). Participants completed practice sessions using these instructions prior in the early stages of the study.

physical motions that cannot practically scale to the context of a real-world user interface. Accordingly, in the fourth and final trial block, participants completed three runs through the complete set of 18 trials (54 total) in an entirely random order. The first run through was discarded as practice, leaving 36 trials per participant for analysis (648 trials in total). This final block enabled us to examine how participants performed the beating gestures after some experience and in a more realistic situation where finger-placement patterns fully vary from trial to trial.

Procedure

The experiment took place in a quiet room with participants seated comfortably at a desk. The experiment started with participants completing a brief demographics questionnaire, reading a set of instructions and then donning the prototype. This was always strapped to the wrist of their non-dominant hand, which was placed palm down on the surface of the desk approximately parallel to their body and in easy view of their eyes and reach of their dominant hand. All touches to the prototype were performed with their dominant hand. The experimental instructions emphasized that they were to perform the beating gestures as fast as possible while maintaining accuracy. Experimenters were available to answer any questions and demonstrated beating gestures in cases where participants had doubts about this issue. Participants then moved on the complete the main study.

Each trial in the study was structured as follows. Firstly, on-screen instructions requested participants to tap the sensors to start. After release, gesture instructions (see Figure 2 for examples) were presented alongside a centrally located green fixation spot. If participants touched the sensors during this period, the fixation spot turned red and the instructions remained on screen indefinitely. After one second without touching the sensors, the fixation spot disappeared and users followed the on-screen instructions to issue the requested beating gesture. Half a second after completing input, users were presented with feedback as to trial correctness (defined in the measures section below). Incomplete or incorrect trials timed-out after 3.5 seconds.

Measures

For each beating gesture we logged the start time of the trial and the time of each touch or release of the sensors. The key measures we extracted from this data were *beating-time*, the time between the first two touches and *release-time*, the time from the release of the first finger until the release of the second in those the nine beating gestures that require it. We also calculated *trial-time*, the time from the first touch until completion of each beating gesture with either a finger placement (for the three gestures that finish with a Hold) or the final finger release (in all other cases).

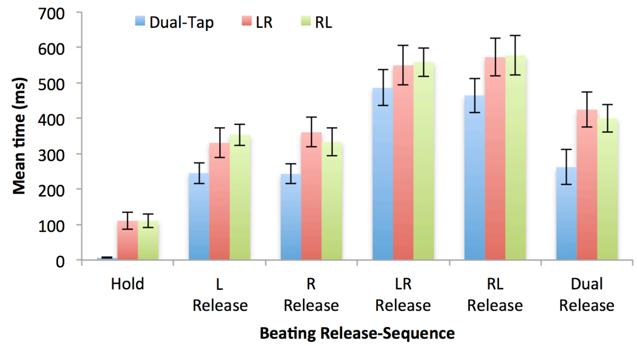

Figure 3. Mean trial-time by placement & release variables.

Errors were also logged based solely on the *sequence of touches* – if users did not produce touches in the order required by each trial within the timeout, the result was logged as an error and the trial returned to the pool still to be presented. There were no constraints, in terms of correctness, related to the timing of touches. For instance, to enter a left-right sequence, the difference between the two touches could be anywhere from 4ms, the lower bound of the touch sensor's temporal resolution, upwards. Equally, dual-tap and dual-release made no constraints on the timing involved in the dual action – both touches or releases could occur either simultaneously or one after the other and any amount of temporal separation was considered valid. This approach ensured we captured every single correctly ordered attempt to produce the beating gestures – thus representing a full picture of how users performed them.

Results

Timing Results

We first report data from the three initial conditions in the study. Overall mean trial-time was short: 355ms (Figure 3). This demonstrates beating gestures can be performed rapidly. Unsurprisingly, the data was affected strongly by the experimental variables – these entail activity that varies in quantity and complexity. A repeated-measures ANOVA, corrected for sphericity violations where appropriate, showed moderate to very strong significant effects of both finger-placement ($F(2, 34) = 12.89$, $\eta_p^2 = 0.43$) and finger release ($F(1.72, 29.1) = 131.42$, $\eta_p^2 = 0.89$), but no interaction between the two variables ($F(3.61, 61.9) = 0.14$, $\eta_p^2 = 0.1$). Post-hoc t-tests with Bonferroni CI adjustments confirmed that Dual-Tap gestures were faster than LR or RL gestures (both $p<0.01$). In terms of the release variable, gestures that did not involve finger releases (e.g. hold) were faster than those with a single finger release which were, in turn, faster than Dual-Release gestures which were, finally, faster than those with two separate finger releases (all at $p = 0.017$ or lower).

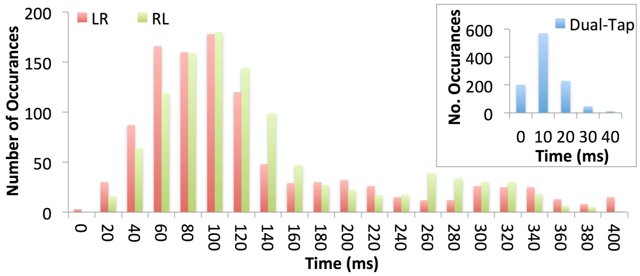

Figure 4. Overall histograms of beating-times for LR, RL and Dual-Tap placement in three main study conditions.

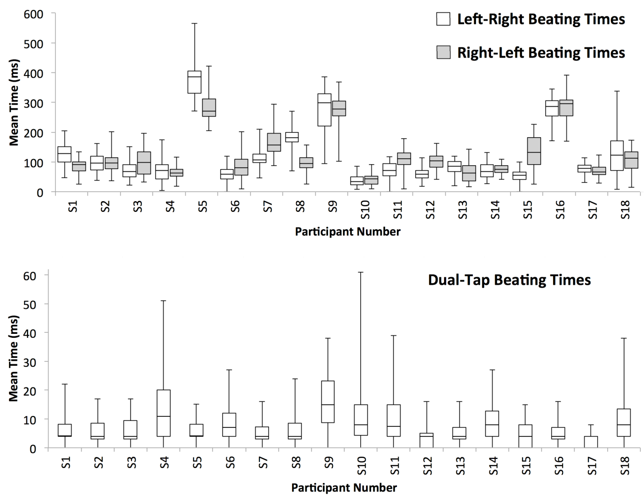

Figure 5. Median beating times shown per participant in three main conditions (outliers removed per participant) for LR and RL (top) and Dual-Tap (bottom) conditions.

More interesting are the beating-time data – the time between two finger contacts. As these distinguish between different beating gestures we examined them descriptively to characterize how distinctly, robustly and reliably they were performed. First, outliers (data in excess of three standard deviations from the mean) for each condition were excluded – a total of 36 trials or 1.1% of data. Beating-time histograms were plotted for the three finger placement sequences (Figure 4). Dual-Taps fell in a relatively narrow distribution and were completed in a mean of 7.33ms (SD 6.88) with none exceeding 35ms. LR and RL taps were completed in means of 121ms (SD 94) and 126ms (SD 82). In contrast to the Dual-Tap data, these plots show a much broader spread of data, including 159 trials (4.96% of total trials) that take place in 35ms or less – overlapping with times from the Dual-Tap condition. There is also a general rightward skew and two peaks in the data: a distinct peak at 100ms and a less prominent peak at 300ms.

To explore how much of this diversity was due to individual differences, we returned to the original dataset, removed outliers on a per participant basis (excluding 33 trials or 1% of data that was over three SD from the mean) and plotted individual box plots (Figure 5). These charts reinforce the generally reliable and consistent performance of the Dual-Tap gesture. However, LR and RL beating

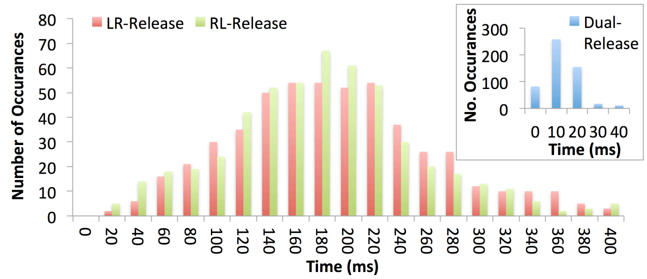

Figure 6. Histograms of release-time for Dual-Release, LR-Release and RL-Release in three main study conditions.

times were more variable. One participant (S10) produced notably more rapid times than others (mean 41ms, SD 19.2), while mean times for the majority fell in the range of 65ms to 150ms and three (S5, S9, S16) produced mean beating times that were two to three times slower – between 270ms and 340ms. Contrasting the charts, we note that two participants with positively skewed Dual-Tap times (S4, S10) also exhibit relatively rapid LR and RL times. Using this data we calculated the per-participant overlap between LR or RL and Dual-Tap input, based on a region three SDs around the Dual-Tap mean. This showed a broadly similar average misclassification rate to the aggregate data – 170 trials (5.2% of overall total) of the RL or LR trials would be misclassified as Dual-Tap. However, 133 of these trials (78%) were due to participants S4 and S10, with the misclassification rate of the remaining 16 participants running at a much lower 1.3%. This analysis suggests that the majority of users can reliably produce distinctive Dual-Tap and LR/RL input, albeit at a range of different speeds. However, a minority of users (11% in this study) exhibit an overlap in performance of these movements, which would make reliable classification of their input challenging.

We analyzed release-time in a similar way. Figure 6 shows a histogram of all data from Dual, LR and RL release trials. This shows trends that broadly mirror those in the finger placement data – Dual-Release is quick and relatively homogenous while LR and RL distributions are much slower and more spread out. Means times on a per-participant basis after outlier removal (20 trials, or 1.2% of data) were as follows: Dual-Release 9.2ms (SD 3.4), LR-Release 191ms (SD 76) and RL-Release 180ms (SD 75). Also on a per participant basis, a total of 37 of the LR/RL trials (2.3% of total trials) fell within three SDs of Dual-Release mean, potentially leading to errors in classifying the beating gestures in this small proportion of trials.

Finally, we characterized performance in the final condition with that in the earlier three. Overall trial time in this block was 313ms (SD 126), a figure a t-test revealed to be significantly lower than that recorded in the initial three conditions (p=0.004). We then examined beating-times. After outlier removal (6 trials or 1% of data), means were: Dual-Tap 7.5ms (SD 3.7), LR 103ms (SD 72) and RL 110ms (SD 69). The overlap between Dual-Tap and LR/RL data was a total of 27 trials (4.1%). Release times (3 outliers, 1% of data) were: Dual-Release 11.5ms (SD 7.7), LR-Release 179ms (SD 63) and RL-Release 173ms (SD 57) with an overlap between Dual and LR/RL of 14 trials (4.3%). Overall, this performance broadly matches that in the first three conditions. This is notable because users performed a more complex task involving random variations of both finger placement and release sequences. This suggests that practice rapidly increased users' fluency with the beating task and they were then able to apply these skills to produce any of the beating gestures at any time.

Error Results

Error rates in the study were low for experimental settings. The mean error rate over the three main conditions was 5.7% (SD 4.0%) and 5.3% (SD 5.0%) in the closing condition, a difference that was not significant (t-test, p=0.44). Given the more challenging nature of the final set of trials, this similarity reinforces the idea that participants rapidly acquired the ability to execute the beating gestures fluently. Given this similarity (and for brevity) we present only an aggregate analysis of errors according to trial type and including data from all four conditions in the study.

We classified errors into two types – *extra*, or those involve making additional undesired input in the form of touches or releases of the sensors following correct completion of the required beating gesture, and *incorrect*, or simply making the wrong input. This data is plotted in Figure 7 by both finger placement (left) and finger release (right) variables and was analyzed using Repeated Measures ANOVA, adjusted for sphericity violations using Greenhouse-Geisser corrections where appropriate and followed up with *post-hoc* t-tests incorporating Bonferroni CI adjustments. The *extra* errors did not vary by finger placement type (F (2, 34) = 0.377, p = 0.68, η_p^2 =0.022), but did vary significantly by finger release type (F (2.86, 48.7) = 9.05, p <0.001, η_p^2 = 0.35) and, with a weak effect size, in the interaction between these variables (F (4.15, 70.52) = 0.377, p = 0.009, η_p^2 =0.18). Due to its small size, the interaction effect was hard to interpret, but the main effect was clearly revealed by the post-hoc analysis: the RL-Release and Dual-Release conditions led to reductions in *extra* errors over, respectively, the Hold (p=0.042), L-Release (p=0.015) and R-Release (p=0.004) and the Hold (p=0.035) and R-Release (p = 0.014) conditions. These results indicate that the majority of *extra* errors were due to early release of the sensors in those conditions where one or both fingers needed to remain in contact. Specifically, this occurred during the 500ms period at the end each trial after input was completed and before feedback was presented. As, in a realistic application, this time period would be minimized, or even removed, we suggest that many of these *extra* errors represent experimental artifacts – participants anticipating when to release the screen to continue to the next trial, and underestimating this time.

More serious are the *incorrect* errors. The RM-ANOVA revealed moderately powerful significant effects of the

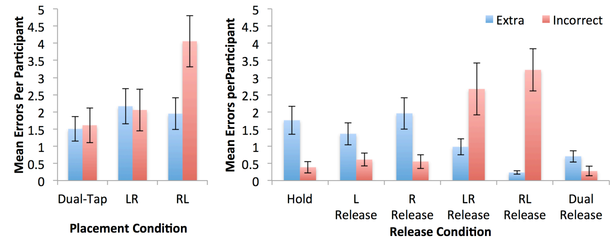

Figure 7. Mean error rates from whole study by finger placement type (left) and finger release type (right).

main effects of finger placement (F (2, 34) = 8.97, p = 0.001, η_p^2 =0.35) and finger release (F (2.04, 34.66) = 12.8, p < 0.001, η_p^2 =0.43) and a weak effect of the interaction between them (F (2.91, 49.42) = 3.38, p = 0.026, η_p^2 =0.17). In terms of finger placement *post-hoc* tests showed Dual-Tap yielded lower errors than RL (p=0.002) while the difference between LR and RL approached significance (p=0.052). Regarding finger release, the only *post-hoc* tests to attain significance involved the RL-Release data – all other release types bar LR-Release resulted in lower errors at either p=0.002 or p=0.003. The interaction plot (not shown) suggests this effect is largely due to a near doubling of *incorrect* errors occurring in trials featuring the specific combination of RL followed by RL-Release. This suggests this particular beating gesture is challenging for users.

Discussion

The major conclusions of this study are that the majority of users performed the beating gestures rapidly (300-400ms) and reliably (approximately 5.5% overall error rate, low for experimental settings). The data compare favorably to other tapping techniques such as Bezel-Tap [16] that exhibit substantially longer task completion times (around 1.5 seconds, possible due to the accelerometer sensing system on which it is based) and larger aggregate errors rates (between 1.75% and 12%). The results also compare well to other forms of smart watch interaction, such as the touch screen radial targeting task studied by Askbrook *et al.* [1]. Although times are not reported, error rates for selecting a target from a set of 12 are stated to be 4.8% (when targets occupy 25% of the screen) – broadly similar to the 5.5% recorded in the production of one beating gesture (from a larger set of 18) recorded here. Similarly, in Harrison and Hudson's [6] discussion of around device interaction, they report that one target can be selected from 22 in around 2 seconds and with a 7.8% error rate, figures that are more than four times slower than those presented here and modestly less reliable. These comparisons provide support for the assertion that the beating gestures are a fast and accurate mechanism for interaction on smart watches.

The data also support recommendations for suitable thresholds for detecting the different beating movements. Dual-Tap and Dual-Release were fast (in the three main conditions, over all users, 28ms at the 99th percentile) indicating 30ms is an appropriate cut-off value. LR and RL were broader distributions with, respectively, figures of 401ms and 372ms at the 99th percentile. This suggests a cut off 400ms would be able to correctly capture these movements. This figure is similar to the data reported in Heo *et al.*'s [9] investigation of sequential distant taps and, as with their work, indicates that beating gestures can be easily distinguished from other forms of input, such as regular single finger taps or double taps (typically spaced at a minimum of 500ms). In this way, we argue that the beating gestures will not interfere with standard watch input in the form of single finger taps, double-taps and strokes. Furthermore, as beating input always starts with two simultaneous on screen touches, we argue it is easy to distinguish from single finger input. We also argue that the thresholds make the system relatively immune to noise – such as unintentional triggers whilst in a pocket. This is partly due to the fact that the beating gestures can rely on existing techniques to ignore on screen input – such as when the screen is turned off, or the watch orientation (as sensed by an accelerometer) is away from norms – and also because it is much more unlikely that two separate but closely timed touches would occur on the small surface of a watch that that a single touch would occur. In this way, we argue that beating input is much less likely to be accidentally triggered than standard single finger input.

However, while these 30ms and 400ms thresholds are valid general recommendations, we also noted considerable individual differences. As such, any realistic beating gestures interface should either allow users to customize thresholds (e.g. as with mouse double-click thresholds) or, adaptively adjust them to match an individual's abilities. This would be particularly important for the minority of users we observed whose data overlapped substantially between Dual-Tap and LR/RL movements – customizing the thresholds would allow these very rapid performers to tweak the system to best match their abilities. It is also worth noting that the challenges in classifying the performance of these users may be due to the study instructions (which emphasized speed) and design of the trials (which did not impose thresholds on performance). As these users logged few sequence errors, we believe they were able performers of the beating gestures who exploited the structure of the study to perform at a pace where noise was inevitable. Future studies of systems with predefined beating thresholds could confirm or refute this idea. An alternative design strategy for these users would be to create interfaces that use only Dual-Tap or LR/RL gestures, but never both, making confounding this input impossible.

The data also show that some beating gestures were harder to perform than others. Specifically, the combining RL with RL-Release led to a significant spike in the error rate, suggesting this pairing should be avoided. There is also, obviously, a clear incentive to use the simplest and shortest beating gestures in interface designs – those that involve fewer strokes are completed more rapidly and with fewer opportunities for error. This point is clearly illustrated in Figure 7 (right) where LR-Release and RL-Release account for a substantial proportion of *incorrect* errors.

BEATS INTERFACE DESIGNS

Encouraged and informed by these results and recommendations, we explored the kinds of interfaces that could be created on small screens using beating gestures. In order to achieve high-resolution multi-touch input and rich, real time graphical feedback in our designs, they were developed using the Processing programming language on a Motorola DROID RAZR XT912. Informal testing with this device indicated it reported touch events with an accuracy of approximately 10ms, sufficient to recognize beating gestures. All touch input and graphical output in the prototypes was restricted to a 30mm square zone (42.4mm or 427 pixels diagonal) in the center of the screen, a typical size for current smartwatch systems such as the Samsung Galaxy Gear (from 41.4mm to 50.1mm diagonal) and the Apple Watch (38.1mm to 43.1mm). The demos used the thresholds for classifying beating gestures derived from the empirical study – pairs of taps needed to take place within 30ms to be classified as Dual input and within 400ms to be classified as LR/RL movements. Taps separated by longer times were not considered as beating input.

Whilst designing the demonstrations we were sensitive to the limited visual space available on smartwatches. Indeed, in order to minimize on-screen clutter we designed systems that were effectively unmarked or invisible until evoked (in much the same way as swipe gestures are currently). To avoid fat-finger problems [16] of users' own bodies obscuring feedback, the designs relied on input (such as strokes) that could be made eyes-free [3] or visuals that were purposefully displaced from a user's on-screen finger [18]. To explore the diversity of the technique we sought to design interactions based on a wide range of beating gestures. Furthermore, we purposefully integrated the beating gestures with traditional input styles such as menus, pie menus and sliders/scrollbars in order to explore how beating gestures could be combined with more established touch screen interactions, one of our key initial design goals. A final design concern was the consequences of inadvertent activation – we attempted to ensure that users would be able to cancel (or reverse) the outcomes of beating input rapidly and easily. In total, we created five prototype applications.

Home Screen – Application Shortcuts. In this application, a typical watch home screen is shown (Figure 8). Touching the screen with two fingers simultaneously (Dual-Tap) causes a three-item menu to appear at the top of the screen, above the likely finger touches. Following up with an LR-Release causes the leftmost command to be activated while a Dual-Release activates the central command and an RL-Release the rightmost command. Although not implemented, stroking off-screen could have been used to cancel the menu. The idea is that users can configure applications to these commands, creating an eyes-free and readily accessible shortcut menu that could be available at all times and from any screen of the watch.

Maps – Tap Menu. In the map application, users can make short strokes to pan the map. To perform further interaction, they can activate a menu with either LR or RL touch sequence followed by releasing the last finger to touch the screen (e.g. RL followed by L-Release). A menu of four buttons then appears in a preset location under their raised finger and users can tap these icons to issue commands such as zoom in or out, or panning to specific locations. This interface is shown in Figure 9. The use of symmetric menus on the right and left of the screen (accessed by LR or RL touches) ensures the same menu is available independent of which arm the watch is worn on. The goal in this design was to provide rapid access to commonly used contextual commands without requiring that they be permanently displayed on the limited available screen space.

Messages – Stroke Menu. Driven by similar motivations, we implemented an alternative symmetric menu design on a watch screen illustrating an incoming text message. In this version, users perform either a LR or RL touch sequence and then raise the first finger that touched the screen (e.g. LR followed by L-Release). A three-quarters pie menu then appears under the finger touching the screen. Stroking towards the top, bottom or center of the screen moves over one of the menu commands and releasing the finger selects this option. In the context of this demo, we showed commands corresponding to reply-to, call-back and send-emoticon – see Figure 10. This UI allowed us to explore the use of beating input followed by simple stroke gestures.

Alarm Clock – Modes and Sliders. The alarm app in Figure 11 shows a simple digital clock. Dual-Tap followed by Dual-Release toggled between the clock and an alarm mode. When in the alarm mode, RL followed by L-Release displayed a slider over the hour digits and away from the on-screen right finger. Moving this finger up and down then adjusted the hours. Similarly, LR followed by R-Release operated similarly for minutes. Together, these enabled a user to configure an alarm time with just a few taps. Regular one finger taps in the alarm mode toggled it on and off. This design showed how beating input could issue direct commands (e.g. mode switch) and also integrate with interfaces for adjusting a pair of analogue variables.

Music Player – Navigation and Sliders. The final app was a simple music player (Figure 12). Visuals included the album artwork, playback position and play/pause state. A regular one-finger tap toggled between play and pause. A centrally positioned volume slider appeared after a Dual-Tap and subsequent vertical strokes caused changes in the displayed volume. To navigate amongst songs, users performed either a LR or RL touch sequence followed by raising the last finger to touch the screen (e.g. RL then L-Release). Subsequent taps with the raised finger moved to the previous or subsequent song, depending on whether the right or left finger was raised. This interface supported eyes free use – although on-screen state feedback was presented, there were no targets or buttons shown.

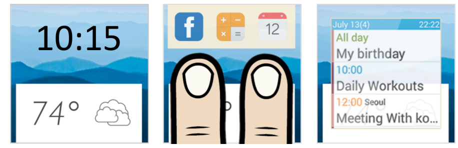

Figure 8. Demo home page (left). Dual-Tap brings up an app shortcut menu (center) and LR-Release, Dual-Release or RL-Release selects the left, center or right command respectively. In the example, RL-Release selects calendar.

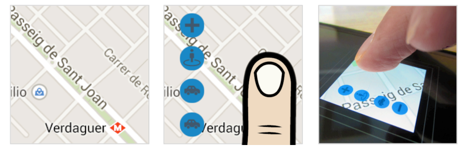

Figure 9. Demo map app (left). RL followed by L-Release shows a menu of options away from the right finger touching the screen (center and right). Subsequent taps with the raised left finger selects items.

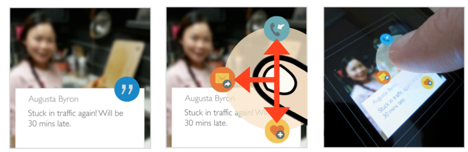

Figure 10. Demo message app (left). LR followed by L-Release shows a menu of options under the right finger touching the screen (center and right). Stroking towards icons and releasing, as in a pie menu, selects commands.

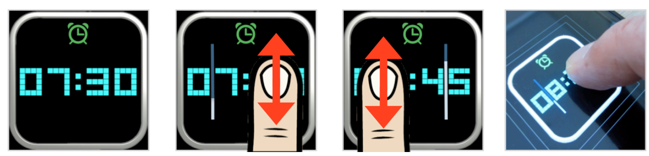

Figure 11. Demo alarm app (left). RL followed by L-Release triggers a slider and vertical strokes set hours (left-center, right). LR followed by R-Release sets minutes in a similar way (right-center). Dual-Tap followed by Dual-Release (not shown) toggles between clock & alarm modes.

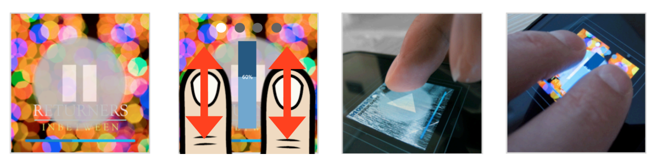

Figure 12. Demo music player (left). Single taps toggle play/pause while Dual-Tap shows a volume slider adjusted by vertical movements (left-center, right). After RL followed by L-Release, subsequent taps with the left finger goes to previous songs (right-center). LR followed by R-Release enables a similar interaction for subsequent songs.

To create a final demonstration system, the five applications were implemented into a single prototype. Swipe navigation, a common technique for moving between pages of content, switched between application screens – upward and downward swipes to navigate between a sequence of the applications and a rightward swipe to return to the first home screen app irrespective of current location.

USER FEEDBACK ON BEATING GESTURES

We ran a user study to gather feedback and assess reactions to these prototypes. The goal of this study was to complement the rigorous quantitative data gathered in the first study with subjective comments and opinions regarding the interaction style when it was instantiated in functioning prototypes. Ten users (five female, a mean of 23 years old) completed the study, all students who stated they had considerable experience of computers (4.8/5), smartphones (5/5) and touchscreens (5/5) but, once again, limited experience with smartwatches (1.3) and other wearable technology (1.3). None had completed the previous study and each was compensated with approximately 10 USD in local currency. The study involved an experimenter demonstrating each application to participants and asking them to comment on the usefulness and easiness of each one. Afterwards, participants tried each system out for themselves. Audio and video (of participants hands and the prototype) were recorded and a second experimenter took live notes. The experiment took approximately 30 minutes per participant.

Results

The prototypes were generally well received. Perhaps the clearest value came in repeated statements relating to the convenience, ease or speed with which the systems could be operated. Commenting on the menu shortcuts on the home screen P3 stated that "usually there are lots of steps to access apps, but not here. Very fast." Similarly P7 liked that the zoom functions on the map were available "right away" and felt the alarm app was much quicker to operate than existing systems. P6 particularly appreciated the music player UI stating it would be useful even on a larger device such as a smartphone.

Users also tended to appreciate the interfaces that employed movements after the beating input. P1 stated that the strokes in the messages app pie menu were "easy" compared to having to "touch some specific icon" and controlling the alarm app was simple because it involved "the finger still on the screen". Similarly, P9 appreciated the "easy manipulation" of the finger placement followed by vertical strokes during setting the alarm time. P10 indicated that the Dual-Tap followed by up/down movements to set volume was "interesting and easy to use".

Conversely, participants reported difficulties in understanding how finger releases could correspond to interface events. The sequential release of both fingers to trigger apps in the home screen was reported by P1 to be "confusing" in how the finger releases related to the on screen content. P6 reiterated this sentiment and stated this movement was "difficult to understand" and that this made it hard to "catch its rhythm" or execute it correctly. Overall these issues were viewed as problems in understanding how input would relate to outcomes rather than any perceived difficulty in physical generating the movements. Those systems, such as the song navigation in the music app, that participants' felt exhibited good mapping (e.g. previous/ next mapping to left/right taps) were seen as simple and effective because, as P9 noted, "the direction of touch and [music] movement was the same".

Users also remarked on the novel nature of the input and the challenges this posed. Discussing various interfaces, P5 and P9 stated that the interface style would "need to be learnt", P7 and P8 that it was "unfamiliar" and P4 worried about the need to refer to a manual. However, these comments were tempered by the perceived simplicity of the input. For example, P6 was initially confused by the experimenter's instructions but interacting with the system "felt ok" after trying it out. Similarly P7 remarked that the message app initially seemed complex but "after a few moments" it was "easy and comfortable" and P9 felt it would be ease to become "accustomed" to the interactions.

An over-arching theme through the comments was that, although the input was easy to perform, its lack of familiarity coupled with an absence of conventional graphical cues meant that users felt it was tricky to initially pick up and might be hard to recall. These comments mirror Norman's sentiments on gestural and natural user interfaces – that a lack of familiar visual affordances negatively impact usability [12]. To address this issue users proposed solutions such as relying on a limited input vocabulary composed a few beating gestures applied systematically over a whole UI, in much the same way that swiping has been generally applied to mobile device interfaces.

Overall, the results of this study suggest were positive as to the major value that the beating gestures provide: quick, easy access to salient commands. However, like most gestural languages, care must be taken to apply the beating gestures consistently so that users can develop a general understanding of how their input will relate and connect to output. Taking advantage of their inherently simple directional (left/right) and temporal (first/second) properties and focusing more on touch than release events represent appropriate ways that designers can achieve this.

CONCLUSIONS

This paper contributes the idea and design of beating gestures: rapid sequences of screen touches and releases made by the index and middle fingers of a users hand. They were conceived as a simple mechanism for increasing the richness of input available via the touch-screens of very small devices such as smart-watches. Beyond this, we also present a user study that characterizes this novel form of gesture, numerically quantifying their performance and concluding they are a rich, rapid and reliable form of input for small screens. Building on these findings we present interaction techniques that integrate beating gestures with archetypical interface tasks such as issuing commands, interacting with menus and adjusting parameters. Our designs attempted to minimize the need for on-screen feedback or ensure clear lines of sight to critical content. A qualitative study of these designs led to generally positive user feedback regarding the convenience and speed of the techniques and concerns about their unfamiliarity. Taken together these outputs and results show the potential of beating gestures as an input technique for small screens.

There are a number of limitations to this work. Firstly, due to the small form factors of the devices, multi-touch screens are not currently common on smartwatch systems. The potential of beating gestures (and other small-screen multi-touch UIs that the community may develop) may change this in the future. Secondly, rapidly polling touch sensors increases power consumption; sensing beating gestures may negatively impact smartwatch battery life. In terms of the empirical work described here, one key, albeit common, problem is the homogeneity of the samples. Further studies that establish whether a broader population of users can readily and reliably perform beating gestures would reinforce the conclusions of this article. A second issue is the lack of formal comparisons against alternative input paradigms. Finally, in the second study, prototypes were implemented on a watch-sized portion of a mobile phone, rather than on a watch, and this lack of ecological validity may have influenced user feedback. In the future, we will develop a fully functioning watch prototype and run comparative tests in mobile scenarios.

Beyond addressing these concerns, future work should consider beating gestures on larger devices where they could be coupled with richer graphical feedback [9, 16], or in specific application areas such as security, where quick and unobtrusive input may have particular value [15]. In sum, we believe the work presented in this paper showcases the potential of beating gestures for increasing the expressivity of touch input on small devices. Furthermore, it achieves this without requiring physical changes to the watch form factor, any fundamentally new sensing hardware (unlike, many prior approaches [13, 15, 19]) or impeding or preventing existing interaction styles based on single taps, strokes, holds and pressure. We believe this demonstrates that beating gestures have the potential to be combined with these established techniques as standard elements of the touch screen interaction paradigm and look forward to exploring the novel interactions this enables.

ACKNOWLEDGEMENTS

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).

REFERENCES

1. Ashbrook, D. Lyons, K. and Starner, T. 2008. An investigation into round touchscreen wristwatch interaction. In MobileHCI '08, 311-314.
2. Blasko, G. and Feiner, S. 2004. An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes. Proc. ISWC'04. IEEE.
3. Brewster, S., Lumsden, J., Bell, M., Hall, M., and Tasker, S. 2003. Multimodal 'eyes-free' interaction techniques for wearable devices. In CHI '03. 473-480.
4. Chen, X., Grossman, T., Wigdor, D.J., and Fitzmaurice, G. 2014. Duet: exploring joint interactions on a smart phone and a smart watch. In Proceedings of CHI '14.
5. Ghomi, E., Faure G., Huot, S., Chapuis, O., and Beaudouin-Lafon, M. 2012. Using rhythmic patterns as an input method. In Procs of CHI '12. 1253-1262.
6. Harrison, C. and Hudson, S.E. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In Proc. UIST '09. 121-124.
7. Harrison C., Schwarz, J., and Hudson, S.E. 2011. TapSense: enhancing finger interaction on touch surfaces. In Proceedings of UIST '11, 627-636.
8. Heo, S. and Lee, G. 2013. Indirect shear force estimation for multi-point shear force operations. In Proceedings of CHI '13. 281-284.
9. Heo, S., Gu, J., and Lee, G. 2014. Expanding touch input vocabulary by using consecutive distant taps. In Proceedings CHI '14. 2597-2606.
10. Hoggan, E., Nacenta, M., Kristensson, P.O., Williamson, J., Oulasvirta, A., and Lehtiö, A. 2013. Multi-touch pinch gestures. In Procs ITS '13. 219-222.
11. Kratz, S. and Rohs, M. 2009. HoverFlow: expanding the design space of around-device interaction. In Proceedings of MobileHCI '09.
12. Norman, D.A. 2010. Natural user interfaces are not natural. interactions 17, 3 (May 2010), 6-10.
13. Oakley, I. and Lee, D. 2014. Interaction on the edge: offset sensing for small devices. In Proceedings CHI '14. ACM, New York, NY, USA, 169-178.
14. Pedersen, E.W. and Hornbæk, K. 2014. Expressive touch: studying tapping force on tabletops. In Proceedings of CHI '14. 421-430.
15. Perrault, S.T., Lecolinet, E., Eagan, J., and Guiard, Y. 2013. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In Procs. of CHI '13.
16. Serrano, M., Lecolinet, E., and Guiard, Y. 2013. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. In Proceedings of CHI '13. 3027-3036.
17. Siek, K.A., Rogers, Y., and Connelly, K.H. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In Proc. INTERACT'05, 267–280.
18. Vogel, D. & Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In Proc. CHI'07, pp. 657-666.
19. Xiao, R., Laput, G. and Harrison, C. 2014. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In Procs. of CHI '14. 193-196.