Faster Command Selection on Touchscreen Watches

Benjamin Lafreniere*, Carl Gutwin†, Andy Cockburn‡, Tovi Grossman*

*Autodesk Research
Toronto, ON, Canada
{ben.lafreniere,tovi.grossman}@autodesk.com

†University of Saskatchewan
Saskatoon, SK, Canada
gutwin@cs.usask.ca

‡University of Canterbury
Christchurch, New Zealand
andy@cosc.canterbury.ac.nz

ABSTRACT

Small touchscreens worn on the wrist are becoming increasingly common, but standard interaction techniques for these devices can be slow, requiring a series of coarse swipes and taps to perform an action. To support faster command selection on watches, we investigate two related interaction techniques that exploit spatial memory. *WristTap* uses multitouch to allow selection in a single action, and *TwoTap* uses a rapid combination of two sequential taps. In three quantitative studies, we investigate the design and performance of these techniques in comparison to standard methods. Results indicate that both techniques are feasible, able to accommodate large numbers of commands, and fast – users are able to quickly learn the techniques and reach performance of ~1.0 seconds per selection, which is approximately one-third of the time of standard commercial techniques. We also provide insights into the types of applications for which these techniques are well-suited, and discuss how the techniques could be extended.

Author Keywords

Smartwatches; interaction techniques; command selection.

ACM Classification Keywords

H.5.2. [User Interfaces]: Interaction styles.

INTRODUCTION

With the introduction of Android Wear and the Apple Watch, wrist-worn touchscreen devices are becoming increasingly common. The promise of these devices is that they enable a faster and more convenient way to access information and perform actions, as compared to alternative devices that take more effort to access (e.g., a smartphone in a pocket). Unfortunately, many of the standard interaction techniques used in current smartwatch platforms are slow and laborious, requiring sequences of coarse swipes and taps on the screen. For example, to select an album in the current version of the Google Music app on Android Wear requires four directional swipes and two taps, just to reach a vertical scrolling list of albums where a selection can be made. This runs counter to prior recommendations that interactions with wearables be kept fast – ideally under four seconds [1].

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858166

The problem arises from the combination of limited input space and limited display output – the small size of the touchscreen has led to a focus on low-bandwidth input techniques such as directional swipe gestures or tapping in lists of items, and the limited display output means that a device cannot show many items at once, necessitating multiple gestures when more than a few targets exist. Higher bandwidth input, such as with voice commands, is one potential solution to this problem, but existing techniques have limitations (e.g., voice control can fail in noisy environments and may not always be socially acceptable).

In this paper we investigate two techniques that exploit spatial memory to enable accelerated command selection on touchscreen watches. The first technique, *WristTap*, is a modification of the FastTap technique for tablets [7], and uses multitouch to speed up selection. Once the user has learned the locations of a menu invocation button and an item, they can use a two-finger touch to perform a selection. The second technique, *TwoTap*, is a variant that works on a similar principle, but uses two sequential touches to select a command. As users learn item locations, they can accelerate the two touches, integrating them into a single "chunked" motor action. In both techniques, the spatially-stable arrangement of commands allows users to build up memory of item locations over time, and use this to accelerate selections.

While previous studies have demonstrated a number of advantages to this kind of spatial interface [7], their benefits are unknown when implemented on tiny displays. We investigate several issues in the design, evaluation, and application of spatial interfaces to touchscreen watches, including:

- How to design spatial menus for a tiny screen, including analysis of item size to support rapid selections;
- How to increase the number of commands by overloading the spatially-stable input space with multiple menus;
- What the learning rates are for spatial techniques, and how overloading item locations affects learning;
- How well the techniques perform in terms of selection time and errors when users are in a standing pose and there are potential distractions present;
- The advantages and disadvantages of the two techniques, and spatial selection on watches more generally.

Our investigation demonstrates that WristTap and TwoTap are feasible on current watch designs, allow for numerous commands, and enable fast selections under realistic conditions. With a small amount of practice, participants achieved selection times of ~0.9 seconds per selection with WristTap, and ~1.0 seconds per selection with TwoTap, compared to ~2.7 seconds per selection for existing techniques based on coarse swipes and taps.

RELATED WORK

Three areas of work influenced our design and analysis – interaction for devices with tiny displays, techniques for rapid command execution, and spatial command interfaces.

Interaction on Tiny Screen Devices and Wearables

Interaction with small touchscreen devices, such as tablets and mobile phones, raises challenges that include the "fat finger" problem [23] of accidental command invocation and target occlusion. These problems are exacerbated on tiny devices, such as watches and wearables, leading to extensive research on interaction methods to overcome them.

Many researchers have examined techniques for tiny devices that eliminate the need to make contact with the display face. Baudisch et al. [4] proposed a variety of methods that use the back of the device, including a watch design that uses the far side of the wrist. While Baudisch et al. examined back-of-device interaction for a wide range of wearable devices, Perrault et al. [18] focused on gestural interaction methods enabled by a multitouch sensing wristband, demonstrating that it can offer an affordable, precise and low power input device. Instead of using the wristband to extend input, Xiao et al. [24] added joystick-based sensing capabilities to a watch bezel, allowing it to sense forces exerted on the watch's case. These forces were then interpreted as 2D panning commands, or as twisting, tilting, and clicking actions. Wrist rotation actions for watch interaction have also been proposed and evaluated [6], and the Apple Watch uses the watch crown as a form of scroll wheel.

A variety of methods have also been proposed to sense interactions near a watch. SkinButtons used infrared sensors to extend the available input surface to include the user's skin on the wrist [15]. Other techniques enable pointing input in the air around a watch using infrared sensors [8] or magnetometers [11]. Finally, speech input is built into both the Android and Apple smartwatch platforms, though it can be impractical in certain social and environmental settings.

Rather than moving interaction off the touchscreen, other researchers have examined methods to improve the effectiveness of touchscreen interaction on tiny displays. Command gestures can be used successfully on small screens [5, 17], though gestures can take time and effort to learn. Ashbrook et al. [2] investigated interactions around the edge of a circular watch, with a focus on error rates. Their method included a semantic zooming technique for interacting with successive levels of hierarchically organised data. More recently, various methods for text entry on watch displays have been investigated, including ZoomBoard [17] and Swipeboard [5]. ZoomBoard uses a series of iterative zooming actions to progressively magnify keys, and Swipeboard uses a series of two directional swiping actions to disambiguate first the keyboard region, and then the target key. Both techniques were designed to exploit familiarity with existing QWERTY layouts, as well as facilitating transitions to expert performance. TouchSense [9] increases touch input vocabulary on tiny displays by using finger-pad contact to infer finger posture during each tap – different meanings can then be assigned to each finger posture at each location.

Our work shares with ZoomBoard and Swipeboard the objective of facilitating transitions to expertise, but we are focused on rapid command invocation rather than text entry, and specifically on the role that spatially-stable menus can play in making this possible.

Rapid Command Execution

Reducing the time taken to invoke commands is an elemental objective in HCI. There are four main categories of methods for doing so: *enabling faster actions*; *reducing the number of actions*; *increasing expressivity per action*; and *enabling parallelism*. There are abundant examples of each, and some key examples follow.

Marking menus [12, 14] *enable faster actions* than standard menu systems. By arranging a series of commands as segments of a 'pie' centered on the cursor, each of the commands can be accessed with a very short movement into the desired segment; a traditional linear menu, in contrast, requires much longer average movement distances.

CommandMaps [20] aim to *reduce the number of actions* required to invoke commands by maximally flattening the command hierarchy – concurrently displaying all commands that would otherwise be contained in menus, tabs and toolbars. Theoretical and empirical results have validated its faster performance as compared to alternative techniques.

Increasing the expressivity of the user's action allows more information to be conveyed without substantially increasing the action's time. This can allow a single action to replace multiple actions, or permit greater flattening of a command hierarchy. For example, Pressure Marks [19] allow sliding gestures to be augmented with pressure information, with example designs including extended marking menus. Apple Watch similarly augments touch with 'Force Touch'.

Finally, *enabling parallelism* allows users to concurrently express their intentions, again allowing more information to be conveyed to the system per unit of time. Familiar forms of parallelism include the use of modifier keys (e.g., shift-click). Multitouch interactions also permit parallel input points, which allow users to express a range of intentions such as two-finger scroll or pinch-to-zoom. Research systems such as multitouch marking menus [16] and finger-count menus [3] have used this capability to let users specify a menu category with the number of finger contacts. Following in this direction, WristTap investigates the potential of using multitouch parallelism to improve command execution on wrist devices.

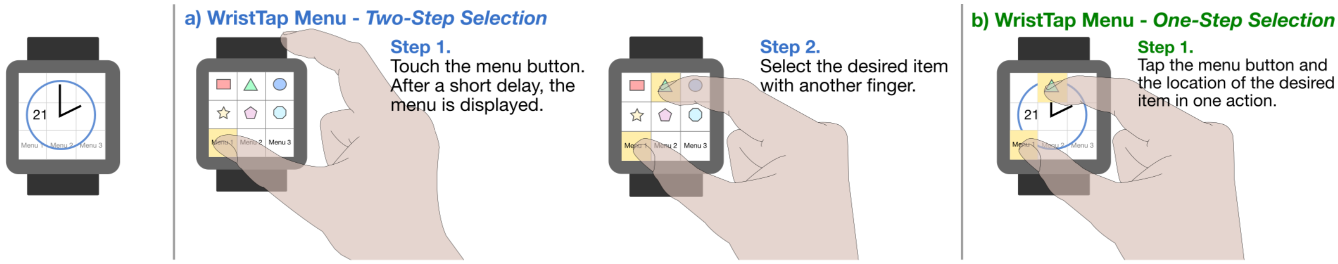

Figure 1. WristTap supports two selection methods: (a) two-step selection, which requires visual search, and (b) one-step selection, which is faster, but requires the user to remember the locations of items.

Spatial Memory Techniques

When interface layouts are spatially stable, users develop knowledge of item locations. This knowledge permits fast interaction because users can quickly recall and anticipate locations rather than conducting a comparatively slow visual search. Many interfaces have been designed to explicitly take advantage of the performance benefits of spatial stability, including marking menus, CommandMaps, and ZoomBoard described earlier (see Scarr et al. [21] for a review of spatial memory in user interfaces).

A key benefit of spatially-stable interfaces is that they can facilitate natural transitions from novice to expert performance. While novices must visually search for items, the physical actions they use to activate them can be more-or-less identical to those used once expert. This idea of facilitating expertise through consistent actions is encapsulated by Kurtenbach's principle that '*guidance should be a physical rehearsal of the way an expert would issue a command'* [13]. Our techniques are designed to embody this principle.

DESIGN OF WRISTTAP AND TWOTAP

To adapt spatially-stable selection techniques to touchscreen watches, we considered five key questions related to: the execution mechanism to enable fast use; the size of command buttons; how to increase the number of commands; placement of menu buttons; and selection feedback. This section describes how we answered these questions for WristTap and TwoTap – similar techniques that differ primarily in the use of multi-touch versus sequential touch, as described below.

How to enable rapid execution?

A spatially-stable arrangement of items allows the user to build up spatial memory, and execute commands by remembering their associated locations rather than searching for the command in the interface. However, to enable this memory-based selection the system must provide an efficient method for the user to communicate a command's location (i.e., an efficient execution mechanism).

For WristTap, we use the execution mechanisms invented for the earlier FastTap technique [7]. Novice users use a *two-step* selection method in which they touch and hold a menu invocation button with a finger or thumb, wait for the menu to appear, then select the desired item with another finger (Figure 1a). Once the user has learned an item's location, they can use a *one-step* selection method in which the menu invocation button and the item location are tapped together, in a single two-finger touch, without waiting for the menu to appear (Figure 1b).

TwoTap, in contrast, works on the principle of simply accelerating a basic interaction to the point where it is driven by spatial memory rather than visual search. TwoTap presents a similar menu organization to WristTap, but instead of a two-finger touch, users touch twice in sequence – once to invoke the menu, and once to select the item (Figure 2). To cancel a selection, the user can tap the invocation button again to close the menu. We hypothesize that as the user becomes familiar with item locations, the two separate taps will be integrated into a single learned motor "chunk", in a similar manner to how double clicking (or entering a lockscreen passcode) are thought of as a single action, rather than a series of separate actions.

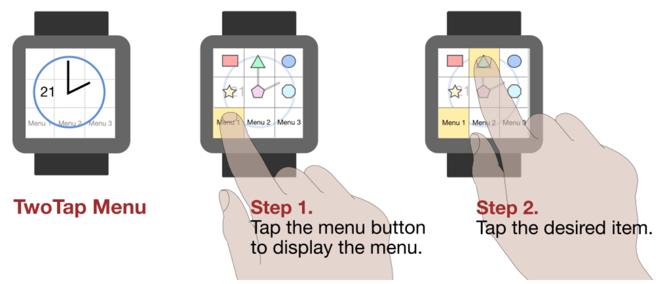

Figure 2. In TwoTap, selections are made with two sequential actions – a tap to open the menu, and another to select an item.

The two mechanisms of WristTap and TwoTap also must be considered in terms of the environments where they will be used. In particular, it is important that the selection mechanism does not conflict with existing actions in the interface – and here WristTap has an advantage over TwoTap. Although many current watch apps make little use of touch interactions on the watch face, it is more likely that an app will want to use single taps (conflicting with TwoTap) than the long-touch and multi-touch actions of WristTap. For similar reasons, WristTap is likely to be more resistant to unintended selections (e.g., from accidental touches to the screen).

How many commands can be used on a tiny screen?

Smartwatches have tiny screens (~4cm diagonal). This limits the number of commands that can be displayed in a spatially-stable presentation, which requires that commands occupy a fixed position on the screen at all times. Previous spatial interfaces have been designed for tablets or PC screens, and their arrangements (e.g., 20 items in a 4×5 grid on a 7-inch screen) are impractical for a watch.

There are two main constraints on the design of a grid menu for a smartwatch: first, each button in the grid must be large enough to be easily selectable with a finger or thumb; second, it must be possible to touch two adjacent buttons simultaneously with two digits (for WristTap). We prototyped several grid sizes and chose three square grid designs for further testing (grids with equal dimensions best matched the near square shape of typical smartwatch screens).

We tested 2×2, 3×3, and 4×4 grids in a performance study that looked at selection time and errors. The results of the study (details below) indicate that a 3×3 grid has the best combination of command-set size and usability – selection times and errors were similar to a 2×2 grid, and were much lower than a 4×4 grid.

How can we increase the number of commands?

A 3×3 grid provides space for nine items, but one location must be used for the menu invocation button (e.g., the lower left grid square), leaving only eight locations for menu items. To increase this number, but still allow the basic mechanisms of both WristTap and TwoTap, our design instead takes the approach of adding additional invocation buttons, each with a different set of command items. The spatial locations of individual items are thus overloaded: there are multiple items at each grid location, and the user selects among these by using different invocation buttons. Figure 3 shows this design as used in one of our studies: there are three invocation buttons, one for each category of items (Colors, Styles, and Shapes), and six items in each category.

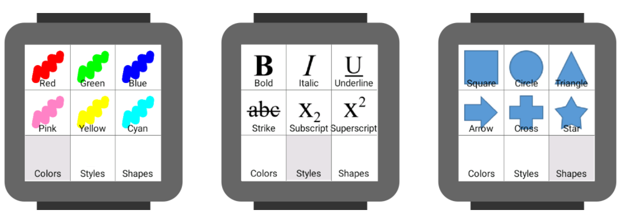

Figure 3. The three-category menu used in Study 2.

Overloading increases the number of items from 8 (1 invocation button + 8 items) to 18 (3 invocation buttons × 6 items per category). The relatively simple nature of applications currently used on smartwatches suggests that 18 items will be enough to cover a wide range of interfaces and use cases. However, with the TwoTap technique, the overloading approach could also be extended to allow nine categories and nine items in each category, for a total of 81 items. (Note that in the current designs we do not mix invocation buttons and item buttons, in order to simplify learning.)

Overloading menus means that the user must now learn two locations for each menu item (invocation button + item location). This could impair performance or lead to mode errors (i.e., selecting the correct location but the wrong category), so we evaluated the idea of spatial overloading in a second study (details below).

Where to place the menu buttons?

Earlier grid menus such as FastTap used a single menu button in the bottom left corner of the grid. While this location works well for tablets, it is unclear whether it would also work on a watch due to occlusion problems and the ergonomics of the required hand orientation. Moreover, modal overloading requires multiple invocation buttons. We built prototypes with the invocation buttons at the top, bottom, left, and right sides of the grid.

From informal testing, we found that using the bottom row of the grid for invocation buttons provided several advantages. For WristTap, the bottom row allows items to be selected using the thumb and index finger in several different finger postures (e.g., a reversed 'C' posture as shown in Figure 1, or a first-and-second-finger posture). For TwoTap, invocation buttons on the bottom row allow the menu to be easily invoked with an index finger without occluding the rest of the interface.

How to provide visual guidance for the technique?

Both WristTap and TwoTap provide four forms of visual guidance to assist selection: invocation marks, menu display, selected item display, and grid marks. First, the techniques display marks (text or icons) on the interface to assist the user in identifying and targeting the invocation buttons. For aesthetic purposes, these marks may be subtly displayed (e.g., Figure 1, left).

Second, after activating an invocation button (with a tap for TwoTap, or a 250ms touch in WristTap), the items for that menu are displayed. The timeout for WristTap is important because displaying the menu is only necessary if the user does not recall the location of an item (and one-step selections can be performed in less than 250ms without opening the menu).

Third, when an item is selected, it is shown in its grid location for 600ms. This feedback provides important confirmation of which item was selected when the user is working from spatial memory rather than visual search.

Fourth, faint transparent grid-marks can be shown to help the user target the different grid locations (see Figure 1, left). Our experience with the two techniques suggests that the 3×3 grid can be easily used without grid-marks – every grid location except for the center is located on an edge or a corner of the screen, which provides natural landmarks for targeting. We test people's ability to make selections without any reference marks at the end of our third study, described below.

Usage contexts

Finally, to demonstrate that our techniques can be integrated with the current interaction paradigms and applications on existing smartwatch platforms, we built two demo applications in the Android Wear platform. Figure 4 (left) shows a music player application with playback controls and a menu of favorite albums. Figure 4 (right) shows a home-screen overlay over the native watch face with toggle buttons for settings, an application launcher, and a quick list of contacts.

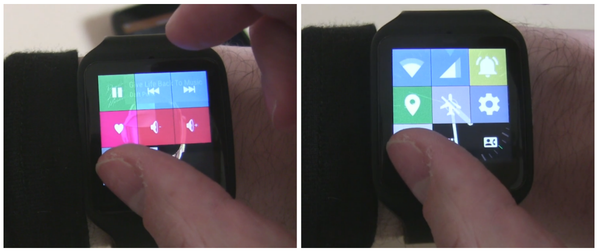

Figure 4. Demo applications: a music player (left), and a home screen application over the native watch face (right).

STUDY 1: GRID SIZE IMPACT ON PERFORMANCE

WristTap and TwoTap both use an arrangement of buttons in a rectangular grid – in two-step WristTap selections and TwoTap selections, grid locations are touched in series, and in one-step WristTap selections, grid locations are touched simultaneously. To understand the human factors of selection in grids on watch-sized screens, we conducted a study of the mechanical aspects of users' capability to acquire targets in various grid dimensions. Specifically, this study was designed to answer the following questions:

- How quickly and accurately can two serial or parallel touches be made in a grid on a touchscreen watch?
- How does performance change over three potential grid sizes (2×2, 3×3, and 4×4) on a 4cm diagonal screen?

To answer the above questions, we conducted a study testing the two selection types in WristTap. We did not explicitly test TwoTap selections, reasoning that one-finger tapping in a grid of buttons is already well-understood, and for the pragmatic purpose of selecting a grid size for our designs, the performance of two-step WristTap selections could act as a lower bound for performance of TwoStep selections.

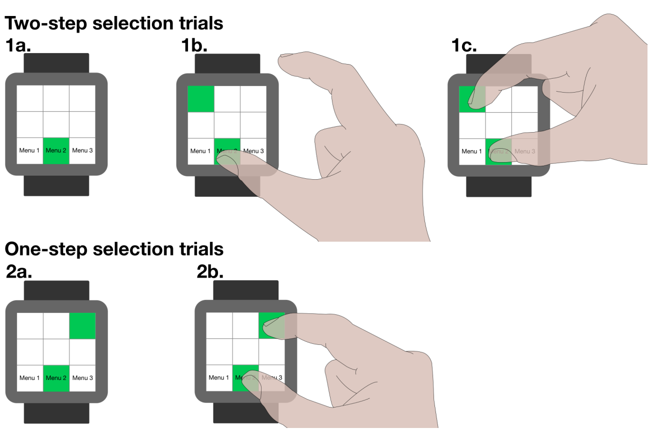

Figure 5. Trials for the two selection types. In two-step selection trials (1a,b,c), the user must first touch the indicated menu button, then select the revealed item. In one-step selection trials (2a,b), the user touches both the menu button and item together.

Tasks and stimulus. The study consisted of a series of trials, each simulating either a two-step or one-step selection in a WristTap menu. For *two-step selection* trials, first a menu button was highlighted in green (Figure 5-1a). The user would touch and hold the indicated menu button with their thumb, causing a target item location to be highlighted in green as well (Figure 5-1b). The participant would then use their index finger to select the target item (Figure 5-1c). These trials simulate novice use of WristTap (opening a menu and selecting a desired item).

For *one-step selection* trials, both a menu button and a target item location were highlighted at the same time (Figure 5-2a). The participant would then tap both locations simultaneously with their thumb and index finger (Figure 5-2b).

Procedure and study design. The study followed a within-subjects design, with each participant experiencing all three grid sizes and both WristTap selection methods. Before starting the study, the two selection methods were explained to the participant, and they performed 12 sample selections using each method in a 4×4 grid. Participants then completed a series of blocks of trials, each consisting of all possible menu and target item locations for one particular grid size (2×2, 3×3, or 4×4) and selection method (two-step or one-step). In each block of trials, all menu and target locations were tested twice, in randomized order, with all menu / target item combinations being tested before any repetition. A one-second delay was enforced between trials. Each participant completed a practice set of blocks for all combinations of grid sizes and item locations, in randomized order, followed by a testing set of blocks for all combinations of grid sizes and item locations.

Selecting an incorrect item, or selecting an item using the incorrect method, were recorded as errors. Errors were notified by vibration through the watch. Participants were instructed to complete trials as quickly and accurately as possible. For each trial, we recorded task completion time, whether an error was made, and data describing individual touches.

Participants. We recruited 14 right-handed participants from a university campus (6 male, 8 female), ages 18-45 (mean 26, SD 9) to take part in this study, and a second study discussed later. Together, the two studies lasted ~60 minutes, and participants were given a $10 honorarium. Participants' average index finger width was 17mm (SD 2mm), and average thumb width was 23mm (SD 3mm). None of the participants reported owning or regularly using a smartwatch. The watch was worn on the participant's left wrist.

Participants performed Studies 1 and 2 in a sitting pose. In Study 3 we evaluate WristTap and TwoTap in a standing pose, with distractions present.

Apparatus. The experiment was conducted on a Sony SmartWatch 3 SWR50 device, with a 4cm-diagonal 320×320-pixel multitouch display (shown in Figure 4). The device ran version 5.0.2 of the Android Wear operating system.

Study 1 – Results

Selection performance by grid size and selection method

Participants' average trial completion times by WristTap selection method and grid size are shown in Figure 6. For the *two-step selection* method, the mean per-participant trial times for the 2×2, 3×3 and 4×4 grid sizes were 1156ms, 1228ms, and 1390ms respectively. *One-step selection*s were substantially faster at 635ms, 657ms, and 805ms respectively. A two-way RM-ANOVA found significant main effects of selection method ($F_{1,13}$=625.5, p < .001, η^2_G = 0.86) and grid size ($F_{2,26}$=34.1, p < .001, η^2_G = 0.37). There was no significant interaction (p = 0.38).

These results indicate the time taken for users to execute the mechanical requirements of WristTap interactions, but they do not provide insight into the user's ability to search for, learn, or recall item locations. Later in the paper, we report on two studies that empirically evaluate WristTap and TwoTap and include these important factors.

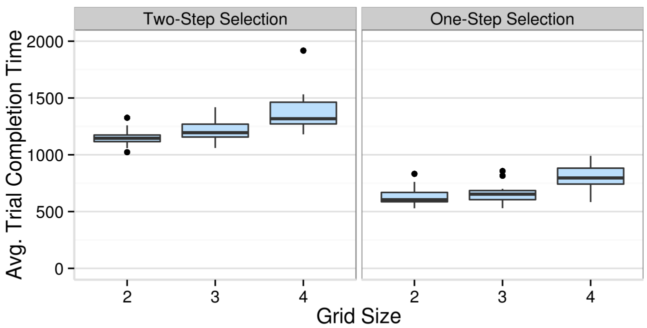

Figure 6. Average participant trial completion times by grid size and selection method.

As a post-hoc investigation, we also looked at differences in selection time by the different finger configurations needed to make the selection. We found that arrangements where the target was directly above the menu button were about 15% faster than those where the target was above and to the left of the menu button (requiring a more extreme rotation of the hand to place the fingers).

Error rates by grid size and selection method

Figure 7 shows box plots of participants' average error rates, by selection method and grid size. For two-step selection, the median error rates for 2×2 and 3×3 grid sizes were both zero (means 0.9% and 1.0% resp.), and for 4×4 it was 3% (mean 4%). For one-step selection, median error rates for the 2×2 and 3×3 grids were also both zero (means 0.9% and 1.4% resp.), and the median error rate for 4×4 was 6% (mean 5%). A two-way RM-ANOVA found a significant main effect of grid size on average error rate ($F_{2,26}$=37.5, p < .001, η^2_G = 0.52), but did not find an effect for selection method. There was no reliable interaction found between grid size and method (p = 0.09).

Overall, the results of this study suggest that a 3×3 grid on a 4cm diagonal screen offers good performance without sacrificing accuracy – two-step WristTap selections can be made in ~1.2 seconds, and one-step WristTap selections in 0.6 seconds. Both have very high touch accuracy (~99%). We can reasonably expect that a 3×3 grid will also work well for TwoTap, since the serial one-finger taps it requires are similar but slightly less complicated than two-step WristTap selections.

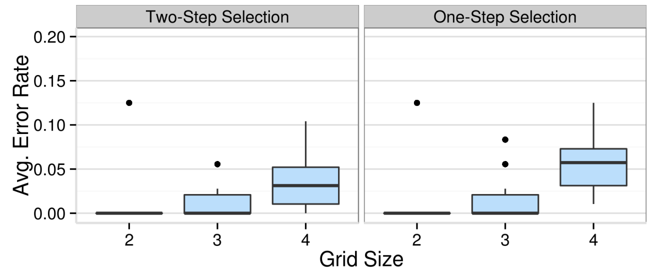

Figure 7. Average participant error rates by grid and method.

STUDY 2: LEARNING AND PERFORMANCE WITH OVERLOADED ITEM LOCATIONS

The previous study examined the mechanical aspects of selecting grid-menu items on a touchscreen watch display, and showed that users can quickly and accurately make selections in a 3×3 grid. Study 2 tests users' ability to learn and use a spatial menu in which individual grid locations are overloaded (e.g., in Figure 3, 'Red', 'Bold' and 'Square' all occupy the top left grid location). Specifically, our goal is to test whether overloading degrades selection performance (e.g., by causing mode errors). Because the overloading issues are similar for both the WristTap and TwoTap techniques, we use only WristTap in this study.

Study 2 – Method

To test the impact of overloading item locations, participants played a game that involved selecting prompted items from the three-category menu shown in Figure 3. To test different levels of overloading, we selected nine items from the menu so that one grid location had three tested items (2-item overloading), one grid location had two tested items (1-item overloading), and the remaining grid locations had one tested item each (no overloading).

Tasks and stimulus. In each trial, a stimulus icon appeared on the screen (Figure 8a). The participant would select the corresponding item from the menu (Figure 8b), and then draw a stroke through the icon to complete the trial (Figure 8c).

The nine target items tested in the study were *Red*, *Yellow*, *Blue*, *Strike*, *Italic*, *Underline*, *Square*, *Triangle*, and *Star*. As mentioned above, this set of items was selected to test the effects of overloading item positions (*Red* and *Square* share a position; *Blue*, *Underline*, and *Triangle* all share a position; and the remaining tested items do not share a position with any other tested item). This set of items was selected manually and was used for all participants.

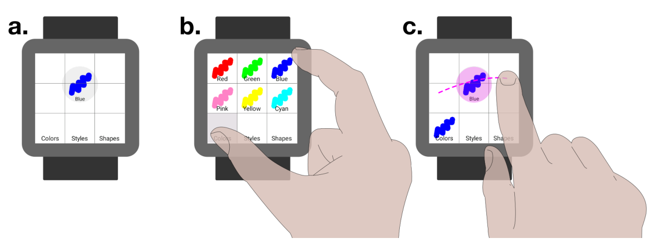

Figure 8. Trials in Study 2. For each trial, an icon appears on the screen (a), the participant selects the corresponding item from the menu (b), and draws a stroke through the icon (c).

Procedure and study design. The study consisted of four stages (three training stages, one testing stage). Each stage consisted of six blocks of nine trials each (one trial for each target item mentioned above, in randomized order).

Since we were primarily interested in the viability and learning cost of overloaded menus, we instructed participants that in the final stage of the study they would be asked to exclusively use one-step selection, and that they should use the three training stages to try and learn this method. Between stages there was a short break of 15 seconds, during which participants were also reminded to try to learn the one-step selection method in anticipation of the final stage.

We logged the time taken to select the correct item for each trial, the selection method used, and the number of errors (selecting an incorrect menu or item) for each trial.

Participants and apparatus. All 14 participants from our first study took part in this study as well, and the study was conducted using the same smartwatch device.

Study 2 – Results

Overloading item locations can affect both selection time and errors: it could increase the time users take to find items (or retrieve their locations from memory); and it could cause mode errors if users select the wrong category. We first consider effects on selection time, and then on errors.

Selection time and overloading

Overall, the median time to select the correct item quickly converges to ~1500ms in the first stage, and then improves only slightly through the remaining stages (Figure 9).

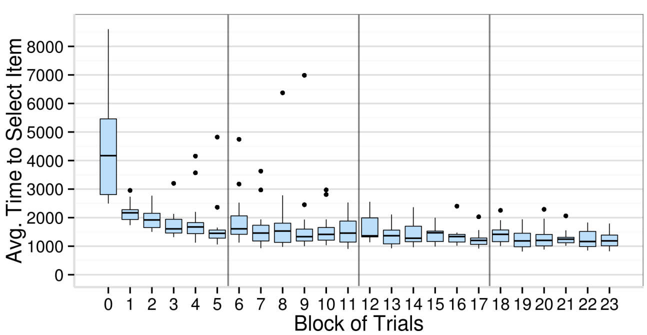

Figure 9. Per-participant average times to select the correct item in each block of nine trials.

To understand the effects of overloading item positions, we looked at the per-participant average times to select the correct item in the final stage of the study, based on the degree of overloading with other tested items. The data showed a slight increase in average time to select the correct item as the number of overlapping items increased, with mean values of 1197ms, 1317ms, and 1413ms for 0, 1, and 2 overlaps respectively. However, a one-way RM-ANOVA failed to show a significant effect ($p = 0.10$).

These findings suggest that the effect of overloading item positions is small (when learning nine items in three 6-item menus). This was reinforced by participants' responses to the question *How difficult was it to transition to using the one-step selection method? (1= Not Very Difficult, 5= Very Difficult)*, where the median response was 1/5, and no-one rated the difficulty higher than 2/5.

Error rates and overloading

We calculated the error rate for each participant during the final stage of the study, and also considered the types of errors that participants made. Data for one participant was eliminated as an outlier, as they had a 31% error rate and the experimenter observed that they had not learned the item positions by the final stage. Overall, the median error rate for users was 3.6%, which was higher than the rate of ~1% observed for one-step selections in a 3×3 grid from Study 1.

Examining participants individually, six of the thirteen had error rates that were consistent with Study 1 (four with zero errors, two with 1.8% errors), and seven had error rates in the 4-10% range. We suspect that this high variation in error rates is due to individual differences in learning item positions in the preceding three stages (i.e., because we instructed participants to use one-step in the final stage of the study, even if that meant guessing at the locations of some items). In a more realistic setting we expect that participants will not use one-step until they are confident in their memory of item locations, and error rates will be lower. However, as a memory-based technique, some errors during one-step selections are inevitable. As we will discuss later, this has design implications for WristTap's use (e.g., with non-destructive command sets or options).

Type of selection	No overloading	1-item overlap	2-item overlap	Overall
Correct (correct menu and position)	314 (99.1%)	156 (94.5%)	233 (93.2%)	703 (95.0%)
Off-by-one error (correct menu, 1 pos. off)	1 (0.3%)	2 (1.2%)	8 (3.2%)	11 (1.5%)
Other error (correct menu, >1 pos. off)	1 (0.3%)	5 (3.0%)	3 (1.2%)	9 (1.2%)
Mode error (incorrect menu, correct pos.)	1 (0.3%)	0	4 (1.6%)	5 (0.7%)
Mode + off-by-one (incorrect menu, 1 pos. off)	0	0	1 (0.4%)	1 (0.1%)
Mode + other (incorrect menu, >1 pos. off)	0	2 (1.2%)	1 (0.4%)	3 (0.4%)

Table 1. Error analysis for one-step selections in the final stage of Study 2.

For further insights, we analyzed the types of errors being made during the final "one-step" stage of Study 2 (Table 1). Overall, most errors occurred when tapping the wrong location within the correct menu (2.7% of selections), while incorrect menu selections (i.e., mode errors) were less common (1.2%). This may be attributable to the visual display of the menu invocation buttons, or to the clear semantic differences between the color, style, and shape categories. Examining trials with 0-, 1-, and 2-item overloading separately, the rate of both mode and non-mode errors increased with greater overloading. This suggests that overloading may have increased the difficulty of learning item locations for the participants with high error rates discussed above.

Transition from two-step to one-step selection

We also examined people's ability to adopt one-step selection in overloaded menus. Figure 10 shows that participants transitioned to one-step selections as the study blocks progressed. Note that the ~100% use of one-step selection in the final stage is due to our study procedure, which asked participants to exclusively use one-step selections in this block (even if doing so caused them to make some errors).

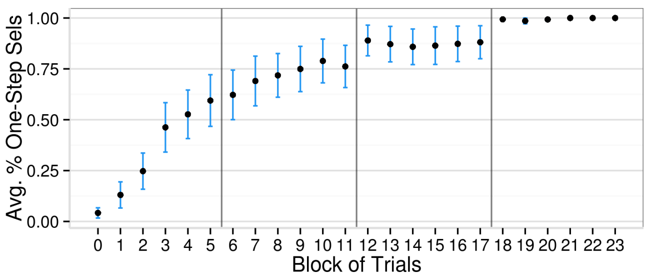

Figure 10. Per-participant average use of the one-step selection method for each block. Error bars indicate standard error.

These results indicate that most participants can quickly learn how to issue one-step selections. However, because we explicitly asked participants to adopt one-step selection, we cannot interpret this as the natural learning rate for the selection method – at most, it should be interpreted as a learning rate for motivated individuals. Studying adoption in realistic settings is a topic for future work.

STUDY 3: EMPIRICAL EVALUATION

Our final goal was to evaluate WristTap and TwoTap in comparison with standard commercial techniques, in a simulation of more realistic conditions. Specifically, we designed Study 3 to answer the following questions:

- How does the selection performance of WristTap and TwoTap compare to standard commercial selection techniques based around coarse swipes and taps?
- How do WristTap and TwoTap perform when the user is in a standing pose, with potential distractions present?

Study 3 – Method

This study tested three menu systems – WristTap, TwoTap, and *Swipe-and-Tap*. Swipe-and-Tap was designed to be broadly representative of selection techniques used on current watches. In Swipe-and-Tap, an item is selected by first using a left swipe to open the category menu (Figure 11b), then selecting a category with a tap to reveal a vertical list of items (Figure 11c), and finally selecting the desired item by scrolling it to the center position and tapping it (the default behavior in Android's WearableListView widget). This is only one potential menu design based around swipes and taps, and at the end of the paper we discuss the performance characteristics of coarse swipes and taps more generally.

Tasks and stimulus. Each participant performed a series of item-selection trials with each of the menu systems. In each trial, a stimulus item was displayed on the screen. To complete the trial, the participant selected the corresponding item. If the participant selected an incorrect item, the watch vibrated and the trial continued until correctly completed.

This study used the same categories and items as Study 2. To control for learning effects, participants were tested on a different 4-item subset for each menu system. The three item sets were (*Red*, *Yellow*, *Italic*, *Star)*, (*Cyan*, *Bold*, *Subscript*, *Circle)*, and (*Green*, *Superscript*, *Square*, *Cross)*.

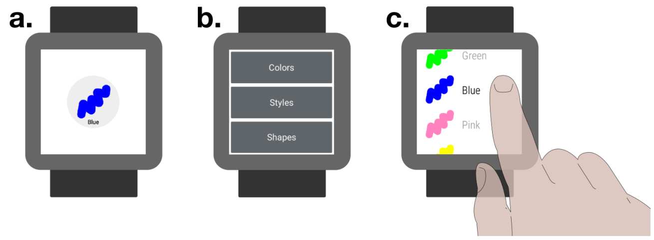

Figure 11. Swipe-and-Tap trials in Study 3. For each trial, an icon appears on the screen (a), the participant left swipes to reveal the category menu and selects a category (b), then selects the item in a vertical list (c).

Procedure and study design. The study used a within-subjects design, with each participant completing a condition for each of the three menu systems. Each condition started with a training stage (18 blocks of 4 items each), followed by a testing stage (6 blocks of 4 items each). To simulate a more realistic environment for smartwatch use, participants performed the testing stage in a standing pose, in front of a screen displaying a point-of-view walking tour of Amsterdam, without sound. The intent was to simulate quick smartwatch interactions while in a neighborhood (e.g., an interaction while waiting for a streetlight to change). During testing blocks, participants started with their arms at their sides. After a 10-second delay, the watch vibrated to indicate the start of a block. The participant raised their arm, tapped a 'Start Block' button on the watch, and performed a block of item selection trials. A text notification on the watch then instructed the participant to lower their arm and wait for the next block. The preceding training stage was performed while sitting, without the tour video playing.

Menu system order was counterbalanced. Item set order was fixed, with all participants experiencing the three item sets in the order presented above. The order of the four items in each block was randomized.

In the WristTap and TwoTap conditions, we included two additional "blind" blocks performed after the testing stage. These were identical to the testing blocks, but participants only saw the stimulus item – the menu system, including grid lines, was invisible. The intent of these blocks was to probe the accuracy of spatial memories formed during interaction with WristTap and TwoTap. For these blocks any selection completed a trial, regardless of errors. We did not include a blind block for Swipe-and-Tap, because its use fundamentally depends on continual visual feedback.

Participants. We recruited 12 right-handed participants from a university campus (5 male, 7 female), ages 17-34 (mean 23, SD 4), none of whom took part in Studies 1 and 2. Two additional participants took part in the study, but were removed from the data analysis for not following experimenter instructions. The studies lasted ~60 minutes, and participants received a $10 honorarium for participating.

Study 3 – Results

Performance of the three techniques

Figure 12 shows the time taken to complete error-free trials in the testing stage of the three conditions. The average trial completion times for WristTap, TwoTap, and Swipe-and-Tap were 948ms, 1020ms, and 2680ms respectively.

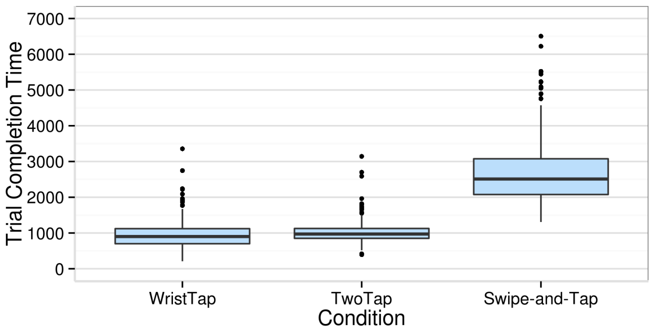

Figure 12. Error-free trial completion times for the testing stage of Study 3, by technique.

A one-way RM-ANOVA found a significant main effect of condition on trial completion time ($F_{2,22}$=154.4, $p < .001$, $\eta^2_G = 0.87$). Post-hoc analysis with pairwise t-tests (Bonferroni corrected) found that both WristTap and TwoTap were significantly faster than Swipe-and-Tap ($p < .001$ in both cases), but did not show a significant difference between WristTap and TwoTap ($p = 0.47$).

Error rates

Table 2 summarizes the percentage of trials per participant in which category errors and selection errors were observed in the testing stage of Study 3.

Error Type	WristTap	TwoTap	Swipe-and-Tap
Selection errors	2.4%	0.3%	0.3%
Category errors	0%	2.8%	0.7%

Table 2. Average per-participant error rates for the testing stage of Study 3.

For selection errors, Swipe-and-Tap and TwoTap were tied for the lowest error rate (0.3%), with WristTap higher at 2.4%. Notably, this is a lower rate of selection errors for WristTap than we saw the final stage of Study 2, which adds credence to our theory that those error rates were artificially high as a result of mandating the use of one-step selection.

A one-way RM-ANOVA found a significant main effect of condition on the rate of selection errors ($F_{2,22}$=4.1, $p < .05$, η^2_G=0.16), but post-hoc pairwise t-tests (Bonferroni-corrected) failed to show significant differences between the individual techniques ($p \geq 0.12$ in all cases).

For category errors, WristTap was lowest with none, followed by Swipe-and-Tap (0.7%) and TwoTap (2.8%). TwoTap's higher rate of category errors and lower rate of selection errors – the reverse of WristTap – suggests that the visual feedback provided by TwoTap enables users to catch errors before they were committed. At the end of the paper, we discuss design issues relating to WristTap and TwoTap's error rates in greater detail.

Spatial ability in WristTap and TwoTap

To understand how well users had developed their spatial and muscle memories for item selections in WristTap and TwoTap, we asked participants to complete two blocks of trials in which selections were completed without any visual information displayed on the watch. With WristTap, 90.6% of trials were completed correctly; and with TwoTap 89.6% of trials were completed correctly. These high success values are important because they suggest that, with relatively little training, participants were able to transition away from relying on strong visual guidance. This suggests the techniques could be used while allowing users to direct some of their attention elsewhere.

Post-study feedback

In the post-study questionnaire, participants were split on whether they preferred WristTap or TwoTap – in response to the question "Which menu system did you prefer for the testing tasks?" 7 of 12 participants indicated TwoTap, 5 of 12 indicated WristTap, and none indicated Swipe-and-Tap or No Preference.

DISCUSSION

Past work has identified smartwatches as an ideal platform for 'microinteractions' – tiny bursts of interaction that take less than four seconds to initiate and complete, allowing the user to maintain attention on the world around them [1]. Our study results suggest that spatially-stable menus can play an important role in enabling microinteractions on touchscreen watches, because they are both easy to learn and fast – Study 2 demonstrated that users can quickly learn three overloaded 3×3 menus, and Study 3 demonstrated that both WristTap and TwoTap can achieve selection times of ~1 second with practice, as compared to ~2.7 seconds for standard techniques. This time savings is significant because a given microinteraction may require multiple selections, or additional actions before or after a selection has been made.

In this section we present some potential explanations for why spatial menus are able to outperform alternative techniques, and discuss the comparative advantages and disadvantages of WristTap and TwoTap.

What makes WristTap and TwoTap faster?

To answer this question, we examined the median time spent on the component actions of selections in error-free testing trials for Study 3 (Figure 13). We observe that the time taken to complete the final stage of a Swipe-and-Tap selection is disproportionately high (median 1209ms) – higher than the time for an entire selection with one-step WristTap or TwoTap, and almost four times the median time to tap one of six spatially-stable items in the second stage of a TwoTap selection (344ms). This suggests that the closed-loop actions required to select one of six items in a scrolling vertical list is a major cause of Swipe-and-Tap's poor performance.

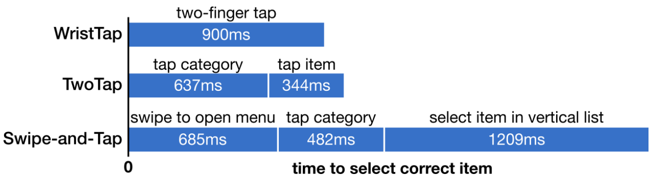

Figure 13. Median time taken for the component actions of selections in the testing stage of Study 3.

Relative merits of WristTap and TwoTap

Study 3 showed that WristTap and TwoTap had fast selection times (with WristTap slightly faster), and that WristTap had lower overall error rates than TwoTap, but a higher percentage of selection errors. Participant preferences were approximately the same for these two techniques. Overall, both techniques have relative strengths that could promote them in different usage situations.

WristTap has potential advantages of a higher performance ceiling and a higher likelihood of integrating with existing interactions. WristTap was fastest overall (even if not significantly so), likely because the simple mechanical cost of carrying out a multitouch tap is less than that of two sequential taps. In addition, as described above in Study 1, certain multi-touch finger configurations were particularly quick – it is possible that WristTap performance could be further improved by restricting the interaction to the fastest finger arrangements. WristTap's other main advantage is that it uses interaction mechanisms that are less likely to be needed for other purposes in a wristwatch app – long-touch and multi-touch can be used on the watch even when taps and swipes are already mapped to behaviors, and it seems reasonable to expect that multi-touch false-positives will be less common than single-touch false positives.

In comparison, TwoTap has the advantage that it is conceptually simpler and has only one selection method, making it easier to learn. It also has a lower error rate, making it well suited for situations where errors might be frustrating or hard to recover from. Additionally, most of the errors with TwoTap occurred during category selection, making them easy to recover from. Another advantage of TwoTap is that it naturally supports a larger number of items – if the entire 3×3 grid was used for menu invocation buttons, the technique could support up to 9×9=81 items (or 8×8=64 items if one item is reserved at each level to cancel the selection). The TwoTap method also naturally extends to deeper hierarchies if access to more commands is necessary. Though these are interesting possibilities, further research is required to whether the motor-memory chunking we observed would be affected by an increase in items and menu levels.

Spatial menus on touchscreen watches

Our study results suggest that selection techniques based on spatial-memory can play an important role on touchscreen watches. In addition to enabling rapid command selection, Study 3 suggests that WristTap and TwoTap require little visual attention to use. This is a particularly useful feature for microinteractions (discussed above) or other scenarios where the user must split their attention between a task on their smartwatch and stimulus in the environment. Comparison with other techniques designed for use with minimal visual attention, such as Bezel Menus [10] and Bezel-Tap gestures [22], is an interesting area for future work.

In some sense, it is surprising that spatial techniques should work well on watch-sized devices at all, considering the display and input space is so limited. The results of Study 2 are important in that they demonstrate that overloaded spatial locations can be used successfully, even on tiny displays.

In addition, technological advances are likely to enable further exploitation of human spatial capabilities on and around watch devices. For example, the entire device – display, bezel, casing, and strap – could potentially be employed to communicate spatially determined information [18, 22], as could spatial gestures around the device [11]. Less dramatically, we expect it will become possible to reduce the width of bezels on smartwatches over time. This is encouraging, because even the smallest models of current touchscreen watches are large enough to support a screen of the size used in our studies, if the bezel width was reduced.

CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated that spatially-stable menus can enable rapid and accurate command selection on touchscreen watch devices. In the process of developing two such selection techniques, we have established the human factors surrounding sequential and parallel touch selections in small 2×2, 3×3, and 4×4 grids on a 4cm-diagonal device; established that users can quickly develop spatial memory for three overlapping six-item grid menus, with minimal cost imposed by the overlapping of item locations; and demonstrated that simple sequential taps in a grid menu are integrated into motor-behavior chunks that allows the user to perform these actions quickly with little visual attention.

In future work, we are interested in exploring whether tactile landmarks indicating grid boundaries could enable spatial memory techniques to be adapted for eyes-free use. We are also interested in exploring spatial menus for watches with circular displays. An interesting issue for a circular form factor is landmarks – in the current designs, the corners of the screen act as natural landmarks, but these will be absent in circular displays. Two prototype designs we are working with are using analogue clock positions (e.g., 3 and 9 o'clock), and using a tiled hexagonal grid.

ACKNOWLEDGMENTS

Thanks to Wayland Bang and Christianne Rooke for their efforts recruiting and running participants, and to our reviewers for their feedback and suggestions. Support for this work was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

1. Daniel L. Ashbrook. 2010. *Enabling Mobile Microinteractions*. PhD thesis, Georgia Institute of Technology.
2. Daniel Ashbrook, Kent Lyons, and Thad Starner. 2008. An Investigation into Round Touchscreen Wristwatch Interaction. *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services* (MobileHCI '08), ACM, 311–314. http://doi.org/10.1145/1409240.1409276
3. Gilles Bailly, Eric Lecolinet, and Yves Guiard. 2010. Finger-count & Radial-stroke Shortcuts: 2 Techniques for Augmenting Linear Menus on Multi-touch Surfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), ACM, 591–594. http://doi.org/10.1145/1753326.1753414
4. Patrick Baudisch and Gerry Chu. 2009. Back-of-device Interaction Allows Creating Very Small Touch Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), ACM, 1923–1932. http://doi.org/10.1145/1518701.1518995
5. Xiang "Anthony" Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14), ACM, 615–620. http://doi.org/10.1145/2642918.2647354
6. Andrew Crossan, John Williamson, Stephen Brewster, and Rod Murray-Smith. 2008. Wrist Rotation for Interaction in Mobile Contexts. *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services* (MobileHCI '08), ACM, 435–438. http://doi.org/10.1145/1409240.1409307
7. Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), ACM, 2617–2626. http://doi.org/10.1145/2556288.2557136
8. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices. *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), ACM, 121–124. http://doi.org/10.1145/1622176.1622199
9. Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, et al. 2014. TouchSense: Expanding Touchscreen Input Vocabulary Using Different Areas of Users' Finger Pads. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), ACM, 189–192. http://doi.org/10.1145/2556288.2557258
10. Mohit Jain and Ravin Balakrishnan. 2012. User Learning and Performance with Bezel Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), ACM, 2221–2230. http://doi.org/10.1145/2207676.2208376
11. Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. 2007. The Gesture Watch: A Wireless Contact-free Gesture Based Wrist Interface. *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers* (ISWC '07), IEEE Computer Society, 1–8. http://doi.org/10.1109/ISWC.2007.4373770
12. Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '94), ACM, 258–264. http://doi.org/10.1145/191666.191759
13. Gordon Paul Kurtenbach. 1993. *The Design and Evaluation of Marking Menus*. PhD thesis, University of Toronto.
14. Gordon P. Kurtenbach, Abigail J. Sellen, and William A. S. Buxton. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Hum.-Comput. Interact.* 8, 1: 1–23. http://doi.org/10.1207/s15327051hci0801_1
15. 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. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14), ACM, 389–394. http://doi.org/10.1145/2642918.2647356
16. G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. 2010. The Design and Evaluation of Multitouch Marking Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), ACM, 2233–2242. http://doi.org/10.1145/1753326.1753663
17. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), ACM, 2799–2802. http://doi.org/10.1145/2470654.2481387
18. Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), ACM, 1451–1460. http://doi.org/10.1145/2470654.2466192
19. Gonzalo A. Ramos and Ravin Balakrishnan. 2007. Pressure Marks. *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (CHI '07), ACM, 1375–1384. http://doi.org/10.1145/1240624.1240834

20. Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving Command Selection with CommandMaps. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), ACM, 257–266. http://doi.org/10.1145/2207676.2207713
21. Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. 2013. Testing the Robustness and Performance of Spatially Consistent Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), ACM, 3139–3148. http://doi.org/10.1145/2470654.2466430
22. Marcos Serrano, Eric Lecolinet, and Yves Guiard. 2013. Bezel-Tap Gestures: Quick Activation of Commands from Sleep Mode on Tablets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), ACM, 3027–3036. http://doi.org/10.1145/2470654.2481421
23. Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT'05), Springer-Verlag, 267–280. http://doi.org/10.1007/11555261_24
24. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), ACM, 193–196. http://doi.org/10.1145/2556288.2557017