PageFlip: Leveraging Page-Flipping Gestures for Efficient Command and Value Selection on Smartwatches

Teng Han[1], Jiannan Li[2], Khalad Hasan[3], Keisuke Nakamura[4], Randy Gomez[4], Ravin Balakrishnan[2], Pourang Irani[1]

[1]University of Manitoba [2]University of Toronto [3]University of Waterloo [4]Honda Research Institute Japan Co., Ltd.

{hanteng, pourang.irani}@cs.umanitoba.ca, {jiannanli, ravin}@dgp.toronto.edu, mkhasan@uwaterloo.ca, {keisuke, r.gomez}@jp.honda-ri.com

ABSTRACT
Selecting an item of interest on smartwatches can be tedious and time-consuming as it involves a series of swipe and tap actions. We present PageFlip, a novel method that combines into a single action multiple touch operations such as command invocation and value selection for efficient interaction on smartwatches. PageFlip operates with a page flip gesture that starts by dragging the UI from a corner of the device. We first design PageFlip by examining its key design factors such as corners, drag directions and drag distances. We next compare PageFlip to a functionally equivalent radial menu and a standard swipe and tap method. Results reveal that PageFlip improves efficiency for both discrete and continuous selection tasks. Finally, we demonstrate novel smartwatch interaction opportunities and a set of applications that can benefit from PageFlip.

Author Keywords
Smartwatches; interaction technique; command and value selection; page-flip gestures.

ACM Classification Keywords
H.5.2. [User Interface]: Interaction styles.

INTRODUCTION
Consumers are becoming increasingly interested in using smartwatches to quickly access information on-the-go. However, in many instances, interactions on these small devices are tedious and time-consuming, requiring minute touch operations such as swiping and tapping while browsing alternatives and selecting an item of interest. For instance, popular apps such as activity tracking, scheduling, and messaging [27] employ sequential operations such as feature invocation followed by value selection. Additionally, fine-grained control like continuous input is not well supported on smartwatches. For instance, Android Wear 2.0 [1] uses a seek bar to adjust media volume which demands precise and accurate touch operations on small screens. Furthermore, switching between UIs leads to an inefficient browsing experience as users have to navigate back and forth between the homescreen and application views.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5620-6/18/04…$15.00
https://doi.org/10.1145/3173574.3174103

Prior work on smartwatch interface design suggests using simple gestures such as a side tap [3, 33], or multiple taps [22, 29] to support rapid command invocation. However, tapping on the small screen is known to be error-prone due to small item sizes [22]. Swiping gestures are shown to be faster than tapping in many instances such as in command invocation [19] and text-entry [7, 34]. However, swiping requires multiple actions and becomes time-consuming as the number of items grow. Additionally, complex gestures such as drawing strokes are limited as they lack established guidelines and feedback [28]. Besides, these work primarily focused on invoking discrete items instead of continuous input, and are therefore only applicable in limited contexts.

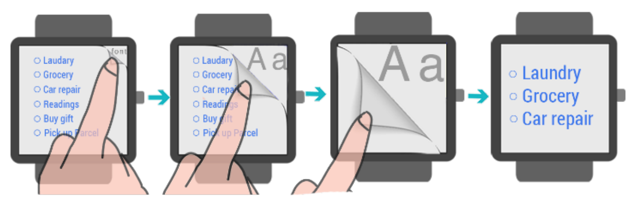

Figure 1. With PageFlip, a user selects a command (e.g. font size) by dragging the top-right corner, and adjusts its values (e.g., text size) by 'peeling' the corner of the page.

In this paper, we present PageFlip, a touchscreen interaction approach that leverages a page-drag gesture to combine command invocation and value selection into a single action. PageFlip gestures are executed via dragging a corner on the screen to different directions and distances. Each corner is mapped to a certain command and can be curled/peeled to browse values available under the command (Figure 1). Such target selection and value adjustment with a single action has been demonstrated on desktop platforms (e.g., marking menu [12, 31]), but never been explored on smartwatches.

PageFlip's novel features offer several advantages. First, it exploits users' spatial memory for corner-command mappings and reduces frequent swiping and tapping actions. Second, it provides real-time and intuitive visual feedback (i.e., curled page effect that maps naturally to users' dragging action) which indicates the current dragging direction and distance from a corner. Third, it supports accessing items in advanced menu layouts where hierarchal or sub-menu items can be accessed via directions and distances. Finally, it allows users to interact with stacked page layouts to make PageFlip applications scalable.

We first investigate several design factors such as corners, discrete angles and distances that can influence users' performance using PageFlip. Based on our results, we refine the PageFlip design and compare it with a functionally equivalent Radial Menu and a standard touchscreen based SwipeTap menu for both discrete and continuous selection tasks. Our results reveal that PageFlip is a promising smartwatch interface for both discrete and continuous selection.

Our contributions include: 1) PageFlip, an interactive method for simultaneous command selection and parameter manipulation on smartwatches; 2) the design and study of suitable parameters for PageFlip; 3) a demonstration of PageFlip's performance over the Radial Menu and SwipeTap menu; and 4) a set of applications demonstrating the unique capabilities of PageFlip over current smartwatch interfaces.

RELATED WORK

We aim to design PageFlip to improve touchscreen input efficiency on smartwatches. We review previous efforts on improving touch-based interfaces and gestural inputs on mobile phones and smartwatches. We also briefly cover existing work on page-flipping user interfaces.

Improving Efficiency of Touch Input

Miniature-sized touchscreens prevent users from efficiently selecting a command from menus and submenus, especially when there are many items. Researchers have proposed using different multi-touch gestures to access items on touchscreen devices. Kin et al. [18] designed a multi-stroke two-handed marking menu for simultaneous menu and sub-menu selections tasks. Benko et al. [4] used two fingers to select small targets. Lepinski et al. [23] designed a marking menu based on simultaneous finger touches. MarkPad [9] used visual and tactile marks to create gestural shortcuts on a touchpad. Similarly, Blaskó et al. [6] proposed to use device corners and edges as tactile landmarks on wearables to support value selections without visual attention. TapSense [15] detects which finger part is used to increase the expressiveness of tapping gestures. ForceDrag [16] used pressure sensing for the same purpose. Pin-and-Cross [25] requires users to uses one finger to pin an object and another finger to select a target from a pre-activated menu. FastTap [13], built on users' spatial memory, is another touch-based interface for rapid access to menu items. A thumb-press on a button displays available items on the screen and an item of interest can be invoked via the index finger while keeping the thumb pressed on the button. A smartwatch version of FastTap was designed in Faster Command [22], where the technique was shown to be faster than standard touch input for command invocation. Their results revealed that multi-touch solutions are promising, but not applicable in many cases due to the miniature sized touch input space.

Other approaches use common finger gestures such as single tap, double tap and dwell. ZoomBoard [30] leveraged a sequence of tap actions that zooms the keyboard with the first tap and selects a key with the second tap. Besides tapping, crossing gestures to select have been shown to be efficient on touch screens [24], but have not been explored for target acquisition on smartwatches. Swipeboard [7] used a sequence of directional swipes to enter text. Both ZoomBoard and Swipeboard work efficiently with a keyboard layout, but have not been applied to general menu selections. Additionally, gestural control usually suffers from lack of established guidelines and feedbacks [28]. Furthermore, previous research mainly focused on discrete target selection and did not explore the feasibility of continuous input on smartwatches.

Supporting Fast and Continuous Input

Marking menus [21] are an efficient discrete command invocation tool. With sequential or multi-stroke gestures, an experienced user can quickly select a target from hierarchical menus. However, the performance of marking menus on small screen platforms (e.g., smartwatches) is an unexplored area. Zhao et al. [43] suggested that for small screens, continuous compound marks may not be suitable due to space constraints. Fast Sliders [26] is a marking menu based approach, designed to support continuous value adjustment by incorporating a slider. Users first flick to a menu item to activate a slider and then drag the mouse to continuously adjust values. ControlMenu [31] and FlowMenu [11, 12] are similar to FastSliders, and support selection and value adjustment in a single drag action via enhanced radial menus. Other approaches that support continuous input on smartwatches include using external or built-in sensors to track finger positions above or around the device [14, 42], wrist rotations [10, 35], or device movements [39, 40].

Page-Flipping Interfaces

Pagination with curled edges and translucent text rendered on the opposite page has a strong visual similarity to e-book readers. The transition, compared with traditional page scrolling, is designed to blend the tactility of real-world pages [38]. On mobile devices, the page-flipping gesture was introduced by the iBook [2], and has been modified in many other commercial apps such as FlipBoard [8]. Bezel-Flipper [17] demonstrated how multi-touch could enhance the design of flipping interfaces for e-books. Beaudouin [5] designed the stacked windows metaphor supporting page rotating and peeling. The page-flipping interface was also used in tangible interactions such as in [36] and a recent work Flippin' [41]. In our work, we extend the use of page-flipping gestures to command invocation on smartwatches, which has never been explored before.

STUDY 1: DESIGNING PAGEFLIP

In the first study, we explored how to design PageFlip as a command invocation mechanism. More specifically, we aimed to support simultaneous command invocation and value selection with PageFlip via hierarchically structured menus and sub-menus. This led to an evaluation of PageFlip's design parameters such as how best to use corners, drag directions and distances from corners for invoking items hierarchically.

Corner: Though all edges and corners could be adopted on PageFlip design, we only considered using the corners as this can be easily distinguished from existing swiping gestures on smartwatches. Accordingly, PageFlip allows a page-drag gesture only from the top-left, top-right, bottom-left and bottom-right corners of the screen.

Angular Segments: Intuitively, a page corner can be dragged to different angular directions to invoke different commands. For instance, a page can be dragged from the top-left corner towards the bottom-left to access a command feature (e.g., font size) whereas moving the curled corner towards the bottom-right corner could be used to invoke another feature (e.g., font face). We divided each corner into 3, 5, and 7 angular segments to identify a suitable number of discrete angular directions i.e., *Angular Segments* that PageFlip can support without affecting users' performance (Figure 2).

Distance Segments: Likewise, the corner can be dragged to different distances and be used to browse discrete or continuous values such as font faces or font sizes. Ideally, a corner tip could at most be moved towards the opposite corner of the touchscreen, revealing half of the second page. Therefore, we decided to include this space to place items. We further divide this distance into 3, 5 and 7 segments where performance differences were expected. This creates a 90° semi-circular layout to place items as shown in Figure 2a and b. The region close to the tip of a corner is extremely small and not ideal for placing more than one menu items. We, therefore, excluded this space in the study. It can be used for higher level commands and previews.

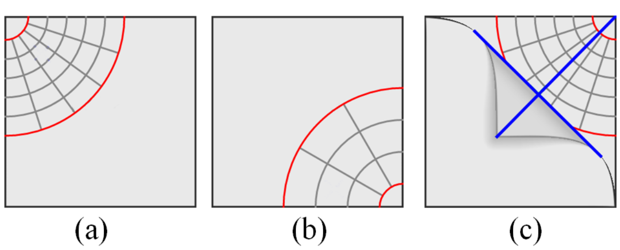

Figure 2. Visualization used to show (a, b) 5 (3) angular segments and 5 (3) distance segments at the top-left (bottom-right) corner. (c) the intersection point between the line from the corner to the flipped tip and the curled edge induces the currently highlighted item.

Selection and visual feedback: PageFlip activates when a user starts dragging a corner of the top page. This action creates a folded page and reveals the visible area on the second page. To show the currently highlighted item, the interface calculates an intersection point of two lines: an imaginary line from the origin corner to the flipped tip and the line created with the curled edge (Figure 2c). Due to indirect input, this visualization reduces finger occlusion and allows precise selection on the small-sized menu items.

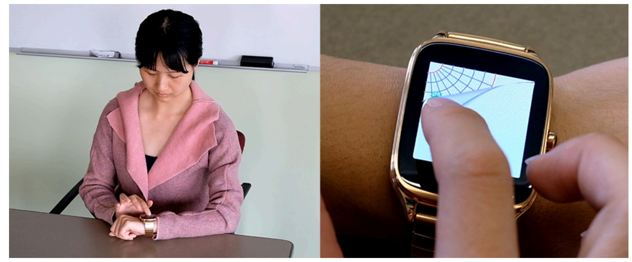

Figure 3. A participant in study 1.

Participants and Apparatus
We recruited 12 participants (3 females, average age 23.2) from a local community. Most participants were in their 20's with a few between 30 and 40 years of age. All of them were right-handed and had no prior experiences of using a smartwatch. We used an Asus ZenWatch-2 (1.63-inch screen, 320 x 320 pixels). PageFlip was implemented using the OpenGL ES framework in an Android Wear application.

Task and Experimental Design
During the experiment, participants were asked to sit in a chair and wear the smartwatch on their non-dominant hand (Figure 3). The watch band was adjusted to align with their natural viewing angle and to fit on their wrists. They were only allowed to use the index finger of the dominant hand to operate on the watch.

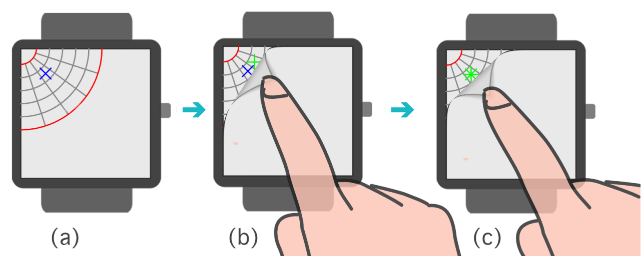

Figure 4. Study 1 design: (a) a target is shown when a trial starts; (b) a user drags the target corner to start selection; (c) the target is successfully selected.

We used a target selection task as shown in Figure 4. A trial starts with a visualization of the next target position on the 90° semi-circular layout (Figure 4a). A participant then starts dragging a corner to flip the top page. This action starts the trial time and unveils the visualization used to display *Angular Segments* and *Distance Segments* (Figure 4b). A blue '×' symbol was used to indicate the target item that the user needs to select. We used a green '+' used as a cursor to represent the currently highlighted item. When the cursor enters the target, the target turns into a green '*' symbol (Figure 4c). With the cursor inside the target region, the participant releases the finger to confirm the selection. A successful selection ends the timer and shows the next trial on the screen. A selection attempt outside the target item is ignored, and the trial continues until the participant selects the target successfully.

We used a 4×3×3 within-subject design for factors *Corner* (top-left, top-right, bottom-left, bottom-right), *Angular Segments* (3, 5, 7) and *Distance Segments* (3, 5, 7). Participants performed 15 repetitions for each condition, yielding a total 540 trials per participant. We gave them practice trials until they felt comfortable with the technique.

All conditions were presented to participants in a random order. The targets were placed randomly in an *Angular Segment* and then in a *Distance Segment*. We instructed the participants to select the target as quickly and accurately as possible. After completing all the trials, we asked them to fill a NASA-TLX form to rate the workload the factors from 1 to 7. The study lasted about 50 mins for each participant.

We recorded the trial time, the time from when a participant starts dragging a corner to the time she successfully selected it. We also calculated error rate by first marking an erroneous trial when the participant failed to select the target. We then divided the total number of erroneous trials by the total number of trials for each condition to get the error rate.

Results

We analyzed trial time using repeated measures ANOVA with Bonferroni corrected pair-wise comparison. We examined error rate using Friedman tests and Wilcoxon tests for pairwise comparison.

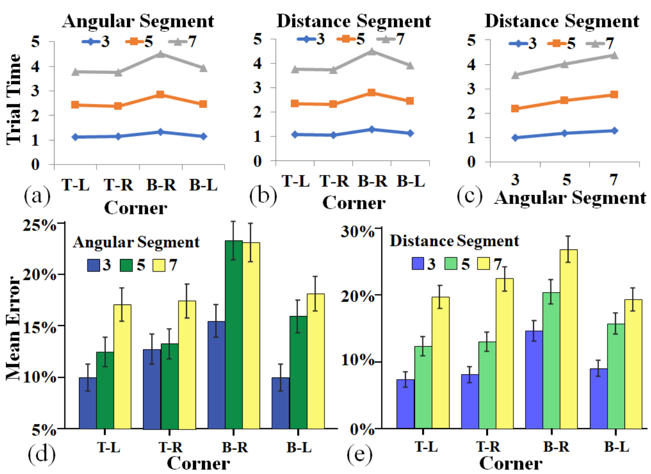

Figure 5. Study 1 results: (a, b, c) average trial time and (d, e) error rates across different conditions. T-L, T-R, B-R and B-L represents Top-Left, Top-Right, Bottom-Right and Bottom-Left, respectively.

Trial Time

The average task completion time (Figure 5) across all conditions was 1.40s. The repeated measure yielded a significant effect of *Corner* ($F_{3,33} = 14.75$, $p < .001$), *Angular Segment* ($F_{2,22} = 74.46$, $p < .001$) and *Distance Segment* ($F_{2,22} = 109.06$, $p < .001$) on trial time, but no significant interaction effect. Post-hoc pairwise comparisons between corners showed that accessing item with the bottom-right corner (M = 1.50s) was significantly slower than with the top-left (M = 1.26s), top-right (M = 1.25s) and bottom-left (M = 1.31s) corners (all $p < .05$). There was no difference between the other corners. Post-hoc pairwise comparisons between *Angular Segments* showed significant difference between each pair (3-segment: M = 1.19s, 5-segment: M = 1.34s, and 7-segment: M = 1.46s) (all $p < 0.001$). We found similar results on *Distance Segments* where we saw significant differences for each pair (3-segments: M = 1.15s, 5-segments: M = 1.33s, 7-segments: M = 1.51s).

Error Rate

There was a statistically significant difference in error rate (Figure 5d and e) based on the corners that the participants used (χ^2 (3, N=12) = 10.3, $p < .05$). Post-hoc pairwise comparisons showed that using the bottom-right corner, with an error rate of 20.7%, was significantly more error-prone than others (top-left: 13.2%, top-right: 14.5%, bottom-left: 14.7%). We didn't find any significant difference for other pairwise comparisons. Our analysis revealed a significant effect of *Angular Segments* on the error rate (χ^2 (2, N=12) = 6, $p < .05$.). Accessing items with 3 segments, with an error rate of 12.1%, was significantly less error-prone than with 5-segments (16.3%) and 7-segments (19.0%). We didn't find any significant difference for other pairwise comparisons. Results also revealed a significant effect for *Distance Segments* (χ^2 (2, N=12) = 20.16, $p < .05$). We observed 9.8%, 15.4% and 22.1% error rate for 3, 5 and 7 *Distance Segments*, respectively, where all pairwise comparisons showed a significant difference.

Subjective Rating

The overall NASA-TLX ratings were less than 3.5. Results were analyzed using a Friedman test with Wilcoxon signed rank tests for pair-wise comparisons. The Friedman test yielded a significant difference in *Corner* (χ^2(3, N=12) = 30.63, $p < .001$), *Angular segments* (χ^2(2, N=12) = 18.43, $p < .001$) and *Distance segments* (χ^2(2, N=12) = 19.16, $p < .001$).

The participants rated the top-left corner as the easiest (M = 1.59) one to access items, followed by the top-right corner (M = 2.17) and bottom-left corner (M = 3.58). The bottom-right corner was rated as the hardest one (M = 5.75) (all $p < .05$). For *Angular Segment*, 3-segments were rated as the easiest (M = 2.08), followed by 5-segments (M = 3.0) and 7-segments (M = 5.0) (all $p < .05$). Similarly, for *Distance Segment*, 3-segments were rated as the easiest (M = 2.0), followed by 5 segments (M = 3.0) and 7-segments (M = 4.42) (all $p < .05$).

Discussion

Our results revealed that selecting an item located at the bottom-right corner required longer trial time and was more error-prone than selecting items located at other corners. This is most likely due to the screen occlusion caused by the right-hand index finger for the right-handed participants that we recruited in the study. Additionally, the result analysis on other factors can be anticipated: trial time and error rates increased with increasing number of angular segments and distance segments. We also observed a higher error rate (e.g., around 15.8%) across the factors. This is primarily caused by the occlusion-prone bottom-right corner as well as higher numbers of angular segments and distance segments that we used in our study. This rate could be reduced by determining suitable design parameters (e.g., the average error rate for top-left, top-right and bottom-left corners with 3 Angular Segment and 3 Distance Segments is 3.53%). Additionally, we suggest using the bottom-right corner for single step tasks such as "next", "cancel" or "reset". As suggested in [37], our results could be mirrored for left-handed participants where the bottom-left corner is likely to be less efficient due to finger occlusion.

STUDY 2: EVALUATING PAGEFLIP

PageFlip leverages users' spatial memory to recall items that are placed in a hierarchical layout using corner-command mappings. It also combines command invocation and value selection into a single drag action. Researchers have presented other techniques such as marking menus [12] or radial menus [26, 31] combined with sliders to support similar tasks. In this study, we compare PageFlip with a functionally equivalent radial menu, as well as a standard smartwatch touch technique using swipe and tap, for both discrete and continuous selection tasks on smartwatches.

Task Type: We explore selection performance of three techniques with two types of tasks: discrete and continuous target selections. Besides the discrete item selection tasks, that we used in our study 1, continuous tasks are often required on smartwatch interactions (e.g., adjusting music volume or changing display brightness). We, therefore, designed 3 discrete tasks: (1) select a letter from a set of letters; (2) select a number from a set of numbers; (3) select an icon from an icon set; and 3 continuous tasks: (1) change the size of a triangle to match with a given triangle; (2) change the color of a filled triangle to match with a target color; and (3) change the stroke weight of a triangle to match a target stroke width.

Technique: We studied the following techniques with the six previously described tasks:

PageFlip: Results from study 1 revealed that using the top-right and top-left corners for accessing items is faster and less error-prone than the other two corners. Therefore, in this study, we included the top-left corner for discrete tasks and top-right corner for continuous tasks. We also considered using 3 angular segments and 5 distance segments as it showed faster trial time (1.14s) with less error rate (8.6%).

With PageFlip, a user first sees a discrete item (e.g., Letter 'E' in Figure 6a) or continuous value (e.g., blue color in Figure 6d) on the screen that they need to select. After reading this, the user can start dragging the top-left or top-right corner to reveal the list of discrete items (Figure 6b) or continuous values (Figure 6e). For discrete tasks, the currently highlighted value is shown with a different color (e.g., green). For continuous tasks, a preview (e.g., dark blue triangle) is included beside the target showing the user's currently acquired continuous value. The user can continue dragging the corner until it matches with the target discrete item or continuous value. When the user believes they have found the item or value (Figure 6c and f), they can lift-off their finger to commit the selection.

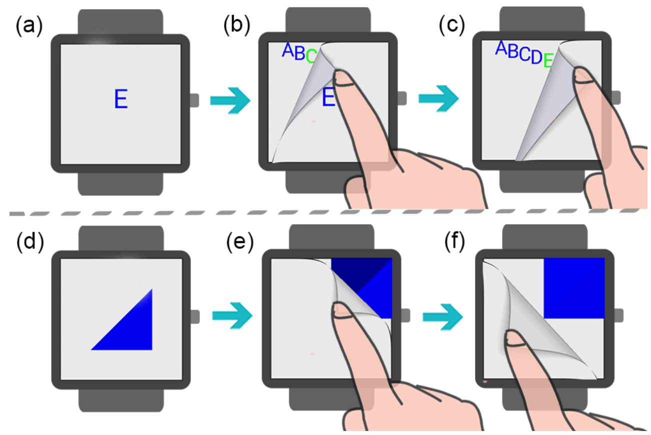

Figure 6. *PageFlip* interface and workflow: (up) select a letter; and (below) change color. Note that we implemented all six tasks, other pictures are omitted to save space.

Radial Menu: We initially considered Control Menus [31], FlowMenus [12] and FaST Sliders [26], which used radial or marking menus [20] for command selection followed by continuous value adjustment. However, our choice was constrained by (i) the small screen size that made scalability and navigating a deep hierarchy difficult [43]; and, (ii) drawing marks as in marking menus without a menu pop-up is likely to conflict with default swipe gestures. To mitigate these challenges, we used a one-level radial menu design plus a slider-based value selection, and embedded a trigger mechanism. We adopted the design from [26], which enabled Radial Menu to be used as a technique for command invocation and continuous value selection.

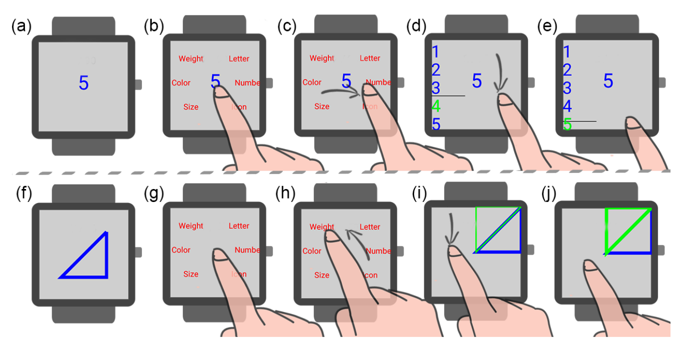

Figure 7. Radial Menu interface and workflow: (up) select a number; and (below) change stroke weight.

In this technique, a user first sees an instruction screen showing a target item or value (Figure 7a and f). Dwell time was used to activate the menu. The user then long-presses on the screen center to trigger a menu where items are arranged in a radial layout around the initial touch position (Figure 7b and g). Menu items for discrete tasks (e.g., letter, number and icon) are displayed on the right side and items for continuous tasks (e.g., weight, color and size) are placed on the left side.

Without releasing the finger, the user can swipe to a direction to select a command (e.g., 'number' and 'weight' in Figure 7c and h, respectively). This action triggers a sub-menu with a list of available options for discrete tasks (Figure 7d) or the target continuous value that the user needs to select (blue triangle in Figure 7i). Like PageFlip, a preview of the currently selected value (green triangle in Figure 7j) is displayed on the screen. To avoid the finger occlusion problem, the sub-menu is shown on the opposite side of the previous swipe direction. That is, swiping right opens the sub-menu to the left and vice-versa. For discrete tasks, a horizontal line is used to indicate the current finger position. An item is highlighted with green color if the line is on top of it (Figure 7e). For continuous tasks, the targets are always placed at the top-right corner. The user adjusts their finger position to match the preview with the target value (Figure 7j). Releasing the finger confirms the selection.

SwipeTap: The standard swipe and tap were included. In this technique, we used the Android Wear 2.0 interface design patterns (e.g., swipe to invoke menus and tap to select) and elements (e.g., seek bar and scroll list).

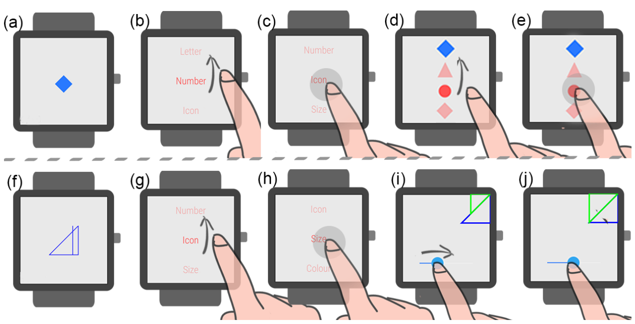

Figure 8. *SwipeTap* interface and workflow: (up) changing size; (below) selecting an icon.

At the beginning of a trial, this technique includes an instruction screen showing the target item or value (Figure 8a and f). A swipe-left gesture reveals a list of scrollable menu items (Figure 8b and g). In both discrete and continuous tasks, the user can tap on an item to access sub-menu items (Figure 8c and h). For the discrete task, they can then scroll the sub-menu items and select an item by tapping on it (Figure 8d and e). For continuous tasks, a target item, a preview and a seek bar are displayed on the screen (Figure 8i). The user can adjust the value by moving the seek bar handle (Figure 8j). When the user believes they have matched the preview with the target, the selection is triggered by a finger release action.

Participants, Task and Experimental Design

We recruited 12 participants (4 females, average age 26.9) for this study. All participants were right-handed and had no prior experience using smartwatches. We used the same apparatus as in study 1.

At the beginning of a trial, the participants saw an instruction screen showing a target either for a discrete or continuous task and the current trial number. We developed an Android app to show experimental conditions. The app started the trial timer when participants tapped on the instruction screen. Participants then performed the command invocation and value selection tasks with one of the three techniques: PageFlip, Radial Menu or SwipeTap. We placed 6 menu items (i.e., Letter, Number, Icon, Weight, Size, Color) for both discrete and continuous tasks. We further included 5 items in each sub-menu (e.g., 'A', 'B', 'C', 'D', 'E' for 'Letter') for discrete tasks. For continuous tasks, the current value was shown based on the distance of curled edge to the corner (PageFlip condition), the finger swipe distance (for Radial Menu*)*, and the seek bar sliding distance (for SwipeTap technique). We stopped the timer for the trial when the participants selected the correct item. Selection attempts in a wrong menu or sub-menu item were ignored. The trial continued until the participants successfully selected the target item or value.

We used a 3×2 within-subjects design for the factors *Technique* (PageFlip, Radial Menu, SwipeTap) and *Task Type* (Discrete, Continuous). Participants were asked to repeat each condition 15 times (target value or item in each trial was randomly determined). In total, we had 270 trials per participant. We counterbalanced *Technique* across participants and randomized the order of *Task Type*.

We recorded trial time for each trial. We further divided the trial time into *Prepare Time*: the time from the trial start to the first touch time, *Menu Activation Time*: from the first touch time to the time when menu appears, *Menu Selection Time*: time involves selecting an item from menu, and *Value Selection Time*: time to select a target value in the sub-menu. As expected, *PageFlip* took no *Menu Activation Time* and *Menu Selection Time* (i.e., dragging a corner to a direction directly triggers a sub-menu for value selection). Two types of error were recorded. A Type 1 Error was logged when participants took a wrong selection attempt to invoke a menu item. A Type 2 Error was registered when participants selected a wrong sub-menu item or value. The participants were asked to complete the tasks as fast and accurately as possible. They were asked to fill a NASA-TLX form for every technique, and their overall preferences (1 – least preferred most and 7 – most preferred). The study session lasted around 40 mins including practice trials.

Results

We used repeated measures ANOVA and Bonferroni corrected paired t-tests for pair-wise comparison to compare the techniques.

Trial Time

We found that participants spent 2.73s, 3.49s and 5.38s on average to complete a successful trial with PageFlip, Radial Menu, and SwipeTap, respectively. A repeated measures ANOVA yielded a significant effect on *Technique* ($F_{2,22}$ = 187.02, $p < .01$) and *Task Type* ($F_{1,11}$ = 246.22, $p < .01$). Post-hoc pairwise comparisons showed a significant difference for each pair (all $p < .01$). We also found that participants were significantly faster with the discrete tasks (M = 3.26s) than with the continuous tasks (M = 4.47s). There was an interaction effect between *Technique* and *Task Type* ($F_{2,22}$ = 102.94, $p < .01$).

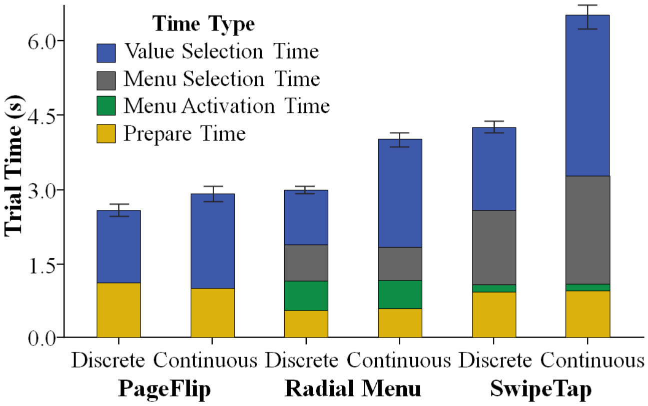

Figure 9. Average trial time for both discrete and continuous tasks across three techniques

We further examined the trial time for discrete and continuous tasks separately. For discrete tasks, the average task completion time for PageFlip, Radial Menu, and SwipeTap were 2.56s, 2.97s and 4.25s, respectively. For continuous tasks, the average task completion time for the three techniques were 2.91s, 4.00s, and 6.51s. A one-way repeated-measures test yielded a significant effect of the techniques on both discrete tasks ($F_{2,22}$ = 106.58, $p < .01$) and continuous tasks ($F_{2,22}$ = 198.44, $p < .01$). For both task categories, each pair of techniques were significantly different (all $p < .05$).

Prepare Time

Prepare Time was significantly shorter with Radial Menu (M = 0.56s) than the other two (both $p < 0.01$). SwipeTap (M = 0.93s) had significantly less Prepare Time than PageFlip (M = 1.05s, p < 0.05). No significant effect of the *TaskType* on Prepare Time was found.

Value Selection Time

We found a significant effect of *Technique* ($F_{2,22}$ = 60.80, $p < .01$), *TaskType* ($F_{1,11}$ = 177.89, $p < .01$) and their interactions ($F_{2,22}$ = 62.52, $p < .01$) for Value Selection Time.

For discrete tasks, a one-way repeated-measure test yielded a significant effect on *Technique* ($F_{2,22}$ = 51.08, $p < .01$). Pairwise comparisons showed that Radial Menu (M = 1.10s) used significantly less time than the others. Additionally, PageFlip (M = 1.46s) required significantly less time than SwipeTap (M = 1.72s) (all $p < .01$). We also found a significant effect on *Technique* ($F_{2,22}$ = 63.68, $p < .01$) for continuous tasks. Pairwise comparison showed that SwipeTap (M = 3.24s) used more time than the others (both $p < .01$), but PageFlip (M = 1.91s) and Radial Menu (M = 2.17s) had no significant difference ($p = .067$).

Error Rate

We used a Friedman test to examine error rate with a Wilcoxon test for pairwise comparison. We analyzed the two types of error separately.

Type 1 Error: We found significant difference between *Techniques* (χ^2 (2, N=12) = 10.67, $p < .01$) and post-hoc pairwise comparisons showed that using PageFlip (0.74%) was significantly less error-prone than Radial Menu (3.70%) and SwipeTap (2.13%). Other pairwise comparisons didn't show any significant difference. Also, we didn't find any significant difference among the *Task Types* (χ^2 (1, N=12) = 1.6, $p = .21$). Discrete (2.28%) and continuous tasks (2.10%) were equally error-prone.

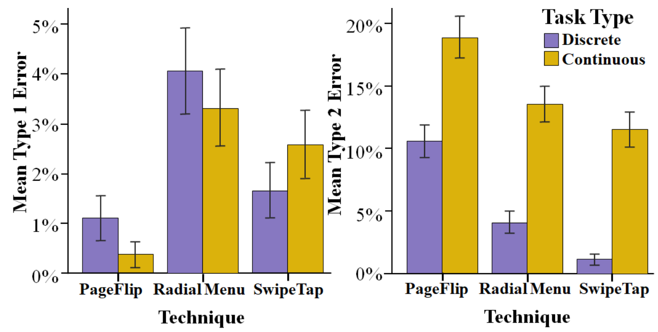

Figure 10: Average Type 1 and Type 2 Error rate on different techniques and tasks.

Type 2 Error: Our results revealed a significant effect for *Technique* (χ^2 (2, N=12) = 16.95, $p < .001$). Post-hoc pairwise comparisons showed that accessing items with PageFlip caused significantly more error (14.72%) than Radial Menu (8.8%) and SwipeTap (6.3%). We didn't find any significant difference for other pairwise comparisons. We also found a significant effect of *Task Type* (χ^2 (1, N=12) = 11.00, $p < .001$) on error rate. Discrete tasks were less error-prone (5.25%) than continuous tasks (14.63%).

Subjective Rating

The overall TLX ratings for *PageFlip* were less than 3.5, while Radial Menu required higher Effort (> 3.5), and SwipeTap required higher Physical Demanding and Effort (both > 3.5). A Friedman test yielded a significant difference in *Technique* ($\chi^2(2)$ = 7.64, $p < .05$). A Wilcoxon signed ranks test found that participants preferred PageFlip (M = 4.41) more than the other two techniques (Radial Menu: 3.17, SwipeTap: 2.08, $p < .05$). No other pairwise comparison was significant.

Discussion

The results indicate that PageFlip is an efficient candidate to be used for designing value selection tasks on smartwatches. PageFlip improves selection time for both discrete and continuous tasks. It has the advantage of having no *menu activation time* and *menu selection time*. Radial Menu had the significantly shortest *prepare time*, which indicates that the participants reacted faster with this technique after seeing the targets. This might due to the finger dwell time that gave them extra time to spatially recall where the menu item was. A 600ms dwell time, that we used based on the system's default settings, caused longer *menu activation times*. We also found that the Radial Menu was faster in selecting discrete items. It leverages the users' finger position to automatically highlight a sub-menu item when the user selects a menu item. For instance, Radial Menu automatically highlights '3' after selecting 'Number' menu as shown in Figure 7. This auto-highlight feature also helps Radial Menu to be faster as users need to travel shorter distances to access other items (e.g., from '3' to '1' or '3' to '5'). Finally, the SwipeTap had longer *menu selection time* and *value selection time* as it required multiple swipe and tap operations to invoke items.

We also observed that PageFlip had less Type 1 Errors, meaning users barely selected a wrong menu item. This is understandable for two reasons. First, discrete and continuous tasks were placed separately in two opposite directions, on the left and right corners, respectively. Second, PageFlip design allows participants to switch fluently between items by changing the angular direction (e.g., from 'E' in 'Letter' to '5' in 'Number'). Such fluent switching between items was not supported by the other techniques. We also acknowledge that our current Radial Menu implementation does not allow cancellation by returning to the center as it could conflict with the 1D sliding operations. Such operations might help reduce Type 1 error, but at the cost of increased trial time.

PageFlip caused more Type 2 Errors. We assumed two reasons. First, discrete items require precise and accurate selection actions as the items were placed in a small semi-circular layout. Second, when selecting continuous values, the preview and target were partially covered by the curled page, making it harder for users to identify the currently obtained value. These visual design drawbacks should be carefully considered to improve the technique's accuracy. Since targets are only visible with curled pages, it is recommended that the menu items should be placed in a more ordered way to foster users' spatial memory and improve selection efficiency. Some participants had difficulty adjusting color or stroke weight to match the targets, regardless of selection technique.

We did not compare the learning effort for these three techniques. It is worth mentioning that for novice users, the current design of PageFlip requires much longer exploration time than the other two techniques as the commands are only visible with curled pages. Additionally, the participants spent a relatively long time in the practice session to get familiar with the menu layouts, especially, for the PageFlip and Radial Menu. This encourages us to analyze the performance of the techniques in future when the users already built spatial mappings of the menus.

APPLICATIONS

In this section, we include three PageFlip applications to demonstrate its possibility for novel smartwatch interactions. These applications utilize one or several unique features that PageFlip supports. For instance, PageFlip could be designed to support a stacked-page layout where items can be placed into multiple layers. Additionally, PageFlip gestures can be used to indicate users' intentions while using smartwatches. For instance, a user can drag a corner to a short distance and hold the finger there to examine a command. A further dragging action can invoke the command or push-back gesture can be used to cancel it. The user may also drag the corner directly to an item if she is familiar with the commands (Figure 11d).

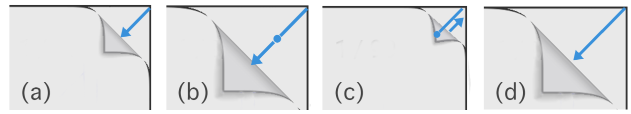

Figure 11. A user (a) starts dragging the corner and holds; (b) continues to drag the corner after holding; (c) pushes back the corner after holding; (d) drags the corner without holding.

Message Edit

Text editing is usually tedious requiring multiple steps of menu invocation and selection process. PageFlip can be used to edit short messages on-the-go. For instance, a user wants to remind her colleagues of a meeting schedule. She first enters "Meeting at 10:30 AM" with her smartwatches' speech input. She is not satisfied with the default font size and color, so she edits the message by dragging the top-right corner to change the font size and color, to make it more visible (Figure 12a). Dragging to one direction picks a color from several options (Figure 12c) and dragging to another direction from the same corner continuously adjusts the font size (Figure 12b). As she finds the changes are not suitable after the first edit, she drags the top-right corner on the second layer to reset it (Figure 12d). When the user is satisfied with the edit, she drags the top-left corner to select a name from the contact list (Figure 12e) and the message is sent upon the finger release. The user is also able to drag the bottom-right corner and flip the whole page (Figure 12f), to create a new message in an efficient way.

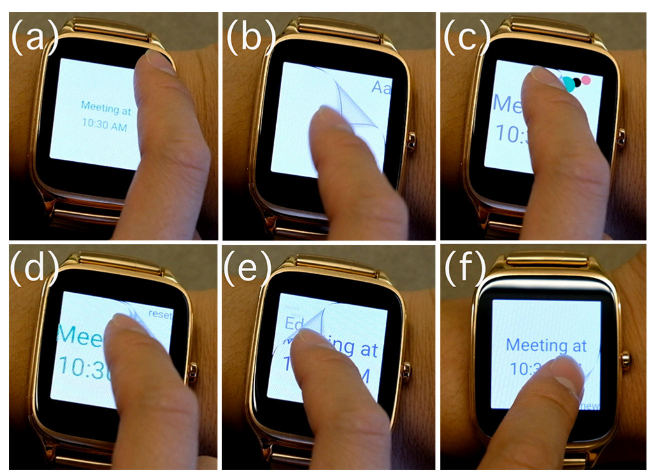

Figure 12. Message editing with PageFlip.

App Notification, Preview and Switching

Displaying an app notification could occupy screen space from users' current task on the smartwatch. PageFlip provides an alternative way for showing notifications and switching to the app. A page corner is curled automatically when there is a notification (Figure 13a). The corner takes less space, thus not affecting users' current task. Meanwhile, the corner can be kept curled to make sure users do not miss it. The user can drag the corner towards the center to preview the notification content (Figure 13b), continue dragging further to open the notification (Figure 13c) or push the corner back to remove the notification (Figure 13d).

Figure 13. App notification, preview and switch with PageFlip.

Pattern Stencil

We designed a pattern stencil technique using PageFlip. We integrate a double click action to demonstrate that PageFlip is compatible with other touchscreen gestures. A user wants to design a stylish letter on her smartwatch for a birthday card. She double-clicks on a pre-selected picture and invokes a "brush" command. She uses her finger to brush on the picture and draws a "Z" (Figure 14a). The user then drags the top-right corner to invoke the "copy" command (Figure 14b). She flips the whole page to the second layer, where the brushed content is already pasted and ready to be used (Figure 14c). This action is similar to tracing on paper or pasting a temporary tattoo, thus easy to learn. Such a technique can also be used to create stylized textured brushes, and cut out a photo.

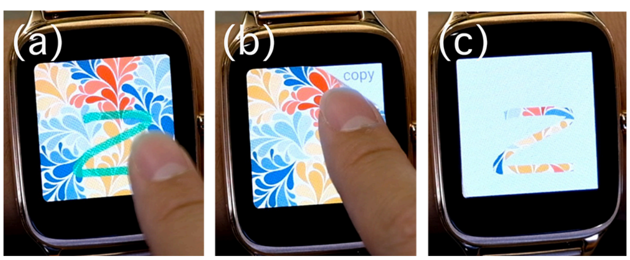

Figure 14. Create a textured letter with PageFlip.

DISCUSSION

We examined three key factors while designing PageFlip in study 1. Results showed that these factors play an influential role in item selection. Our findings also suggest excluding the bottom-right corner to place items as well as using limited number items in both angular and distance segments. We also observed that participants frequently overshoot when the targets were placed close to the corner. This is primarily due to the small area allocated to the inner circles in the semi-circular layout. Further experimentation is needed to find suitable layout design for PageFlip.

Results from our second study clearly indicate that, in comparison to Radial Menu and SwipeTap, PageFlip reduces command invocation and value selection time. This is primarily due to the integration of multiple operations (i.e., command invocation and value selection) into a single action as well as the benefits of having no menu activation time. We also found that PageFlip requires more time in value selections. Further design exploration is required to reduce this time. As the menu items are only partially visible with the flipped corners, we observed overshooting with PageFlip. We anticipate that including an overview of all the available options could help users to eliminate this problem. Besides, an ordered list could implicitly help users quickly navigating to a desired item.

Usability Challenges

Interactions with PageFlip have Discoverability, Affordance and Learnability challenges, especially for novice users. Like most gestural input on touchscreens, it is hard for novice users to find available gestures or commands. It is essential to design a tutorial mode for novice users. For instance, a corner could automatically be curled to show available or recommended commands upon the task context and users' actions. Further strategies could help novices learn quickly. For example, consistent command-corner mappings for frequent commands or categorizing discrete and continuous commands into left and right corners, as in study 2, could be used to facilitate spatial memory and improve learning. Using many angular dragging directions with one corner might increase users' mental and physical effort. Although participants are familiar with the page-flipping gestures of other metaphors (e.g., books), the use of the PageFlip for command invocation and value selection are new to them. UI designers could consider incorporating PageFlip gestures as a complementary feature to existing touch input, which could make such page-flipping gestures common to users.

The studies were carried out in an ideal environment where users were seated in case they got tired. Smartwatches are often used in mobile contexts. It requires us to further explore the performance of using PageFlip while users are walking or standing.

In the demo applications, we only show PageFlip to navigate to the next page with a page flipping gesture, but not the previous. Such actions are easy on large screens as two pages can be displayed side by side (e.g., iBook on iPad). Such actions could be accomplished with a back-flip gesture.

Round vs. Square Face

Smartwatches are commonly manufactured with round- and square-faced design [32]. PageFlip is designed based on corner flipping gestures for two reasons. First, curled corners can be used as an anchor, indicating users' dragging direction and distance. Second, it is easy to distinguish PageFlip from existing swiping gestures (e.g., swiping up to bring up the notifications). These corner-flipping features are not available on round faces. However, round faces have their own uniqueness, and could potentially support flipping gestures from any direction by incorporating virtual corners.

Future Work

Future work includes exploring the design parameters and performance in more realistic mobile contexts, exploring corner-command mappings that can better leverage users' spatial memory and be more intuitive. It is also important to carry out a study that investigates users' learning effort to use PageFlip with realistic tasks. Designing PageFlip on round screen watches is also worth exploring. The abstract tasks used in our studies helped investigate users' motor capabilities. While they do not capture the full range of real tasks possible with PageFlip, they present the limits of PageFlip and allow the design of real tasks, as in our demo applications. The challenge remains defining better visual representations for larger labels. PageFlip supports multi-layer stacked page layout, thus facilitating more menu items. Another approach is to leverage the fisheye effect to allow interacting with a larger number of smaller menu items. These designs demand further investigations.

CONCLUSION

In this paper, we explored the design and performance of PageFlip, a technique that leverages corner-command mappings and supports command invocation and value selection in a single corner-drag action on smartwatches. We first examined the design parameters such as corners, angular segments and distance segments, and then compared the performance of PageFlip with a standard swipe-tap method and a functionally equivalent radial menu. The results indicated that PageFlip significantly increased the efficiency for both discrete and continuous tasks by combining multiple operations into a single action. Finally, we used three applications to demonstrate suitable uses of PageFlip for novel smartwatch interaction.

ACKNOWLEDGEMENT

This research was partially funded by the Honda Research Institute Japan Co., Ltd. and the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

1. Android Wear 2.0. Retrieved September 14, 2017 from https://developer.android.com/wear/index.html
2. Apple Developer. Human Interface Guidelines. Retrieved September 14, 2017 from https://developer.apple.com/ios/human-interface-guidelines/views/pages/
3. Daniel Ashbrook, Kent Lyons, and Thad Starner. 2008. An investigation into round touchscreen wristwatch interaction. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services* (MobileHCI '08). ACM, New York, NY, USA, 311-314. http://dx.doi.org/10.1145/1409240.1409276
4. Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, NY, USA, 1263-1272. http://dx.doi.org/10.1145/1124772.1124963
5. Michel Beaudouin-Lafon. 2001. Novel interaction techniques for overlapping windows. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '01). ACM, New York, NY, USA, 153-154. http://dx.doi.org/10.1145/502348.502371
6. Gábor Blaskó, Steven Feiner. 2006. Evaluation of an Eyes-Free Cursorless Numeric Entry System for Wearable Computers. *10th IEEE International Symposium on Wearable Computers* (ISWC 2006), Montreux, Switzerland. pp.21-28. doi: 10.1109/ISWC.2006.286338
7. Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). ACM, New York, NY, USA, 615-620. https://doi-org/10.1145/2642918.2647354
8. Flipboard. Retrieved September 14, 2017 from https://flipboard.com/
9. Bruno Fruchard, Eric Lecolinet, and Olivier Chapuis. 2017. MarkPad: Augmenting Touchpads for Command Selection. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 5630-5642. https://doi-org/10.1145/3025453.3025486
10. Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. WristWhirl: One-handed Continuous Smartwatch Input using Wrist Gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16). ACM, New York, NY, USA, 861-872. https://doi-org/10.1145/2984511.2984563
11. François Guimbretière, Andrew Martin, and Terry Winograd. 2005. Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.* 12, 3 (September 2005), 460-476. http://dx.doi.org/10.1145/1096737.1096742
12. François Guimbretiére and Terry Winograd. 2000. FlowMenu: combining command, text, and data entry. In Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST '00). ACM, New York, NY, USA, 213-216. http://dx.doi.org/10.1145/354401.354778
13. Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster command selection on tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 2617-2626. http://dx.doi.org/10.1145/2556288.2557136
14. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on*

User interface software and technology (UIST '09). ACM, New York, NY, USA, 121-124. https://doi-org/10.1145/1622176.1622199

15. Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11). ACM, New York, NY, USA, 627-636. https://doi-org/10.1145/2047196.2047279
16. Seongkook Heo and Geehyuk Lee. 2012. ForceDrag: using pressure as a touch input modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (OzCHI '12), Vivienne Farrell, Graham Farrell, Caslon Chua, Weidong Huang, Raj Vasa, and Clinton Woodward (Eds.). ACM, New York, NY, USA, 204-207. http://dx.doi.org/10.1145/2414536.2414572
17. Sangtae Kim, Jaejeung Kim, and Soobin Lee. 2013. Bezel-flipper: design of a light-weight flipping interface for e-books. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13). ACM, New York, NY, USA, 1719-1724. https://doi-org/10.1145/2468356.2468664
18. Kenrick Kin, Björn Hartmann, and Maneesh Agrawala. 2011. Two-handed marking menus for multitouch devices. *ACM Trans. Comput.-Hum. Interact*. 18, 3, Article 16 (August 2011), 23 pages. https://doi-org/10.1145/1993060.1993066
19. Yuki Kubo, Buntarou Shizuki, and Jiro Tanaka. 2016. B2B-Swipe: Swipe Gesture for Rectangular Smartwatches from a Bezel to a Bezel. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 3852-3856. https://doi-org/10.1145/2858036.2858216
20. Gordon Kurtenbatch. 1993. The Design and Evaluation of Marking Menus. Thesis. University of Toronto.
21. Gordon Kurtenbach and William Buxton. 1993. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (CHI '93). ACM, New York, NY, USA, 482-487. http://dx.doi.org/10.1145/169059.169426
22. Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. 2016. Faster Command Selection on Touchscreen Watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 4663-4674. https://doi-org/10.1145/2858036.2858166
23. G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. 2010. The design and evaluation of multitouch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10). ACM, New York, NY, USA, 2233-2242. https://doi.org/10.1145/1753326.1753663
24. Yuexing Luo and Daniel Vogel. 2014. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 2627-2636. http://dx.doi.org/10.1145/2556288.2557397
25. Yuexing Luo and Daniel Vogel. 2015. Pin-and-Cross: A Unimanual Multitouch Technique Combining Static Touches with Crossing Selection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). ACM, New York, NY, USA, 323-332. https://doi-org/10.1145/2807442.2807444
26. Michael McGuffin, Nicolas Burtnyk, and Gord Kurtenbach. 2002. FaST Sliders: Integrating Marking Menus and the Adjustment of Continuous Values. In GI 2002 Conference proceedings: Graphics Interface Conference. pp. 35-42.
27. Chulhong Min, Seungwoo Kang, Chungkuk Yoo, Jeehoon Cha, Sangwon Choi, Younghan Oh, and Junehwa Song. 2015. Exploring current practices for battery use and management of smartwatches. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers* (ISWC '15). ACM, New York, NY, USA, 11-18. https://doi.org/10.1145/2802083.2802085
28. Donald A. Norman and Jakob Nielsen. 2010. Gestural interfaces: a step backward in usability. interactions 17, 5 (September 2010), 46-49. https://doi.org/10.1145/1836216.1836228
29. Ian Oakley, DoYoung Lee, MD. Rasel Islam, and Augusto Esteves. 2015. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 1237-1246. https://doi-org/10.1145/2702123.2702226
30. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 2799-2802. DOI: https://doi-org/10.1145/2470654.2481387
31. Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. 2000. Control menus: excecution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '00). ACM, New York, NY, USA, 263-264. http://dx.doi.org/10.1145/633292.633446
32. Round v square faced smartwatches: We ask the experts which is best. Retrieved September 14, 2017 from https://www.wareable.com/smartwatches/round-v-square-smartwatches-which-is-best

33. Marcos Serrano, Eric Lecolinet, and Yves Guiard. 2013. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 3027-3036. https://doi-org/10.1145/2470654.2481421
34. Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu, and Mike Chen. 2016. SwipeKey: a swipe-based keyboard design for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16). ACM, New York, NY, USA, 60-71. https://doi.org/10.1145/2935334.2935336
35. Ke Sun, Yuntao Wang, Chun Yu, Yukang Yan, Hongyi Wen, and Yuanchun Shi. 2017. Float: One-Handed and Touch-Free Target Selection on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 692-704. https://doi-org/10.1145/3025453.3026027
36. Taichi Tajika, Tomoko Yonezawa, and Noriaki Mitsunaga. 2008. Intuitive page-turning interface of e-books on flexible e-paper based on user studies. In *Proceedings of the 16th ACM international conference on Multimedia* (MM '08). ACM, New York, NY, USA, 793-796. https://doi.org/10.1145/1459359.1459489
37. Daniel Vogel and Géry Casiez. 2012. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 2307-2316. http://dx.doi.org/10.1145/2207676.2208390
38. Why Flipping Through Paper-Like Pages Endures in The Digital World. 2012. Retrieved September 14, 2017 from https://www.wired.com/2012/05/why-flipping-through-paper-like-pages-endures-in-the-digital-world/
39. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 193-196. https://doi-org/10.1145/2556288.2557017
40. Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, and Aaron Quigley. 2016. Sidetap & Slingshot Gestures on Unmodified Smartwatches. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16 Adjunct). ACM, New York, NY, USA, 189-190. https://doi-org/10.1145/2984751.2984763
41. Koichi Yoshino, Koichi Obata, and Satoru Tokuhisa. 2017. FLIPPIN': Exploring a Paper-based Book UI Design in a Public Space. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 1508-1517. https://doi.org/10.1145/3025453.3025981e
42. Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 1491-1503. https://doi-org/10.1145/2858036.2858082
43. Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (UIST '04). ACM, New York, NY, USA, 33-42. http://dx.doi.org/10.1145/1029632.1029639