# Exploring Callout Design in Selection Task for Ultra-small Touch Screen Devices

**Akira Ishii and Buntarou Shizuki**
University of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan
{ishii,shizuki}@iplab.cs.tsukuba.ac.jp

## ABSTRACT

Ultra-small touch screen devices tend to suffer from occlusion or the *fat finger problem* owing to their limited input area. A callout could solve these problems by displaying a copy of the occluded area in a non-occluded area. However, callout designs for ultra-small touch screen devices have not yet been explored in depth. In this study, we chose three design factors (each factor has two levels) from various factors and conducted an experiment to examine eight callout designs in the selection task for ultra-small touch screen devices. The results of our experiment matched the results from previous research; however, we also obtained results unique to ultra-small devices. The results showed that the selection speed was higher when the content of the callout was changed continuously, the error rate decreased when the content of the callout was changed continuously and a pointer was displayed to indicate the touched position within the callout, and the workload decreased when the content was changed continuously. Further, as a design factor, the position of the callout would not affect the selection performance.

### Author Keywords

Interaction technique; occlusion; fat finger; small target acquisition; pointing; wearable device; smartwatch.

### ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g., HCI): User Interfaces – Input devices and strategies, Interaction styles, Screen design; D.2.2 Design Tools and Techniques: User interfaces.

## INTRODUCTION

Ultra-small touch screen devices (henceforth referred to as ultra-small devices), such as smartwatches, must be small because they are worn on the body; thus, the touch screen of these devices is ultra-small. Owing to their limited input area, ultra-small devices are more prone to occlusion or the *fat finger problem* (Siek et al., 2005) than smartphones or slate devices, as shown in Figure 1a. Therefore, the applications of these ultra-small devices are limited to the display of information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

OzCHI '16, November 29 – December 2 2016, Launceston, TAS, Australia
Copyright © 2016 ACM. ISBN 978-1-4503-4618-4/16/11…$15.00
http://dx.doi.org/10.1145/3010915.3010922.

A promising solution to the problem would be the use of a callout. Traditionally, a callout has been used for a pull-quote or text extract that is typically placed apart from an article in a larger or contrasting font in an article. In the field of human-computer interaction (HCI), a callout has been used to support various selection tasks (e.g., menu selection, map, and text entry) on a touch screen (Leiva et al., 2015; Vogel and Balakrishnan, 2010; Vogel and Baudisch, 2007), including a magnifying glass in iOS and Android for text selection; a callout displays a copy of the occluded area in a non-occluded area, thus eliminating the occlusion caused by the finger, as shown in Figure 1bc. Owing to the severely limited input area of ultra-small devices, a carefully designed callout is likely to resolve the problem of occlusion, thus improving the usability of a selection task on ultra-small devices.

However, while various design factors could affect the performance of a callout, callout designs for ultra-small devices have not yet been explored in depth. In an experiment to compare the performance of three QWERTY-based soft keyboards on ultra-small devices, Leiva et al. (Leiva et al., 2015) examined a callout design factor that determines whether only the selected item is displayed (*single* mode) or the selected item and its surroundings are displayed (*surroundings* mode). The results of the experiment indicated that the single mode yielded a worse text-entry speed and error rate. In our previous study, we evaluated three callout design factors for ultra-small devices (Ishii et al., 2016): Presentation Method (Continuous and Discrete), Presentation Position (Fixed and Following), and Pointer Existence (NotExisting and Existing).

The objective of this study is to extract design guidelines for callouts on ultra-small devices. Therefore, we examined the three factors from our previous study (Ishii et al., 2016) by using a larger number of participants and analyzed the results. The results of our experiment matched the results of the previous study; in addition, we also obtained slightly different results that were unique to ultra-small devices, thus yielding a different set of guidelines.

## RELATED WORK

### Small Target Acquisition

Numerous studies have investigated small target acquisition and solutions to the finger occlusion problem (Chapuis and Dragicevic, 2011). Among them, several studies have explored the performance of touch selection for different input techniques and different input devices (e.g., Cockburn et al., 2012; MacKenzie et al., 1991). However, such studies have used devices – such as smartphones and slate devices – that are larger than ultra-small devices.

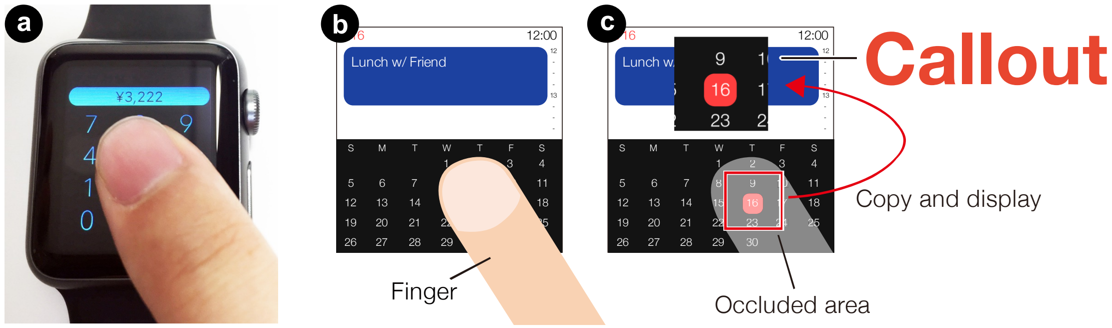


**Figure 1. a) Ultra-small devices are more prone to occlusion. b, c) A callout definition in this study. b) During a touch operation by a finger on the screen of an ultra-small touch screen device, most of the screen is occluded by the finger. c) A callout is used to display a copy of the occluded area, thus eliminating the occlusion caused by the finger.**

Numerous studies have investigated small target acquisition with specialized hardware. NanoStylus (Xia et al., 2015) uses a finger-mounted fine-tip stylus to reduce the occlusion problem. NanoTouch (Baudisch and Chu, 2009) addresses the problem of finger occlusion by using touch input at the back of the device. Holz and Baudisch (Holz and Baudisch, 2010) showed that touch accuracy could be improved by using individual finger postures and a user ID, both of which were deduced by using specialized hardware to sense the fingerprints. While these studies showed that a specific device or additional specialized hardware successfully improved the performance of small target acquisition, our study focuses on solving the finger occlusion problem by employing only a software technique that uses a callout.

### Text Entry on Ultra-small Devices

Text entry typically involves frequent small target acquisition (i.e., selecting a key); therefore, studies focusing on text entry on ultra-small devices investigate the same problem as our study. In ZoomBoard (Oney et al., 2013) and Swipeboard (Chen et al., 2014), touch gestures on the touch screen trigger iterative zooming (visual magnification) until a certain level of zoom is reached; thus, text can be entered using a QWERTY keyboard on ultra-small devices. WatchWriter (Gordon et al., 2016) employs touch and gesture typing with statistical decoding to enable text entry on ultra-small devices. Komninos and Dunlop (Komninos and Dunlop, 2014) used a specialized keyboard with six large keys and adopted alternative or next-word predictions based on a dictionary to enable text entry on ultra-small devices. SplitBoard (Hong et al., 2015) splits a QWERTY keyboard into two parts, and thus, increases the size of each key to enable text entry on ultra-small devices. In contrast with these studies based on gestures, zooming, or specialized design of a display layout, a callout technique for small target acquisition could be an alternative approach that can be generally applied to text entry and the selection task.

### Occlusion-aware Interfaces

The occlusion problem can be avoided by using occlusion-aware interfaces (e.g., Brandl et al., 2009; Khalilbeigi et al., 2012; Vogel and Balakrishnan, 2010). However, these techniques were designed for tabletop or slate devices, and therefore, would not be suitable for ultra-small devices whose screen size is too small for the use of such occlusion-aware interfaces.

### Offset Cursor Technique

Occlusion and ambiguity in selection can be avoided by using the Offset Cursor technique (Potter et al., 1988; Sears and Shneiderman, 1991). In this technique, a pointer is displayed at a fixed distance above the touch point; this pointer serves as a software version of a stylus. However, a disadvantage of this technique is that a user cannot touch the item directly. In contrast, an advantage of the callout technique is that it allows the user to directly access any screen area (Vogel and Baudisch, 2007).

### Callout Technique

The studies that are most related to ours are (Leiva et al., 2015; Vogel and Baudisch, 2007).

Shift (Vogel and Baudisch, 2007) is a target acquisition technique that uses a callout on a personal digital assistant (PDA). In order to eliminate occlusion caused by a finger, Shift uses a callout that shows a copy of the area occluded by the finger in a non-occluded area. In the study, the following design factors were explored: placement, shape, zooming ratio, and control-display (CD) ratio.

Leiva et al. (Leiva et al., 2015) use a callout to enable text entry using a QWERTY keyboard on ultra-small devices. The study focused on proposing a technique for text entry and on comparing related techniques to determine the most appropriate one. The authors examined one callout design factor for text entry; this factor determines whether only the selected item is displayed (*single* mode) or the selected item and its surroundings are displayed (*surroundings* mode). The results of the experiment indicated that the single mode yielded a worse text-entry speed and error rate. In contrast with the research by Leiva et al., our study focuses on the comparison among different callout designs to evaluate their effectiveness for the selection task on ultra-small devices. Therefore, we examined other callout design factors for ultra-small devices.

## CALLOUT DESIGN IN SELECTION

As described in the previous section, the various callout design factors forming a large design space are: placement, shape, and zooming ratio (Vogel and Baudisch, 2007); and a factor that determines the size of the area displayed by a callout (Leiva et al., 2015). However, to determine a good callout design in a scenario wherein a user selects a tiny target (e.g., small icon or keyboard) on an ultra-small device, we chose the three factors in (Ishii et al., 2016) – i.e., Presentation Method, Presentation Position, and

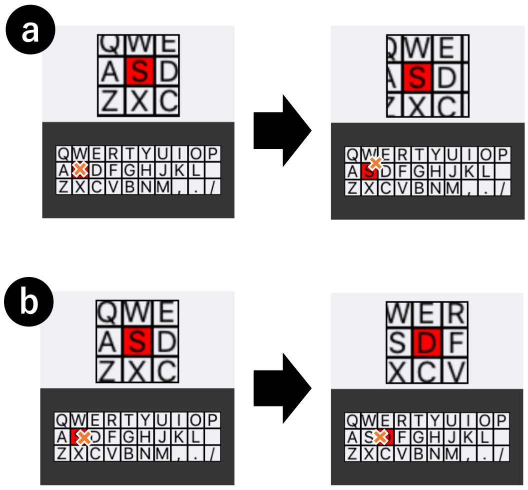


**Figure 2. Presentation Method: a) Continuous and b) Discrete. The X mark indicates the position of the finger of the user.**

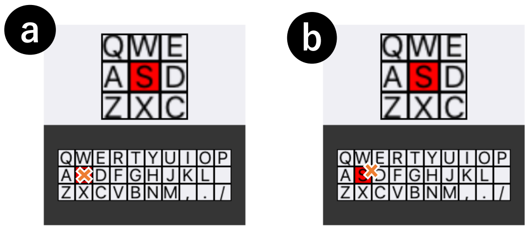


**Figure 3. In Discrete, even if the finger is moved from the position in a) to the position in b), the callout displays the same content because the finger stays on the same item.**

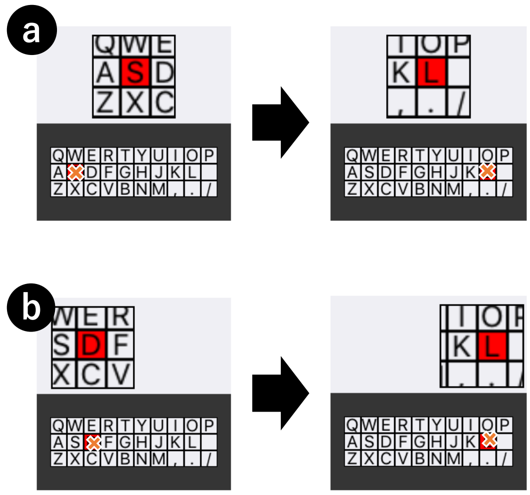


**Figure 4. Presentation Position: a) Fixed and b) Following.**

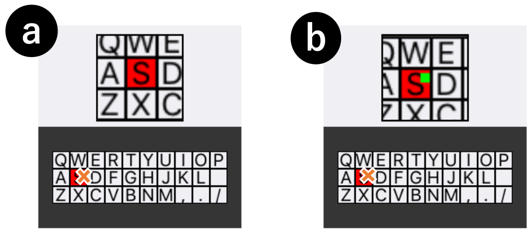


**Figure 5. Pointer Existence: a) NotExisting and b) Existing. The green dot indicates the position of the touch point.**

Pointer Existence – in the selection task as another step to explore the large design space. If we could find a callout design that performs well in this scenario, a user could use various applications (e.g., instant messenger, micro-blogging) on ultra-small devices instead of using notification-style applications; thus, the user could tap on an arbitrary location and type text more easily.

### Presentation Method

*Presentation Method* is the factor that determines how the content of a callout changes in response to a user operation; it has two levels: Continuous and Discrete.

In Continuous, the content of a callout is changed continuously in response to the current touch point, as shown in Figure 2a. The area around the position touched by a finger is directly displayed on the callout. Shift (Vogel and Baudisch, 2007) and the standard copy-and-paste operation in iOS adopt this approach.

In Discrete, the content of a callout is changed discretely in response to the current selection of an item by a finger, as shown in Figures 2b and 3. Unlike Continuous, the content is not changed as long as the finger stays on the same item, even if the finger is moved. When another item is selected as the user moves the finger, the content is changed to display the surrounding area of the newly selected item. This approach is similar to the callout of the software keyboard in iOS. The content is changed whenever an item is selected; therefore, the change could serve as visual feedback to notify the user that an item is selected.

### Presentation Position

*Presentation Position* is the factor that determines how the position of a callout changes in response to a user operation; it has two levels: Fixed and Following.

In Fixed, the position of the callout is fixed at the center, as shown in Figure 4a. In this condition, the user observes the same area within the display during operation because the callout position is fixed. Therefore, this condition has the advantage of low gaze movement.

In Following, the position of the callout follows the finger position, as shown in Figure 4b. As a result, the user would receive the visual feedback of their operation in the form of the change in the position of the callout.

### Pointer Existence

*Pointer Existence* is the factor that determines whether the actual touch point of the user is displayed on a callout as a pointer (Existing) or not (NotExisting), as shown in Figure 5. If the pointer is displayed, the user could recognize the actual touched point.

## EXPERIMENTAL EXPLORATION

We conducted an experiment to explore the usability of various combinations of the three factors of callout design. In this experiment, participants performed a target selection task – i.e., the selection of tiny targets – in all the factorial combinations of the three factors. We recorded all the actions of the participants during the experiment and analyzed the actions according to the following three criteria: selection speed, error rate, and workload.

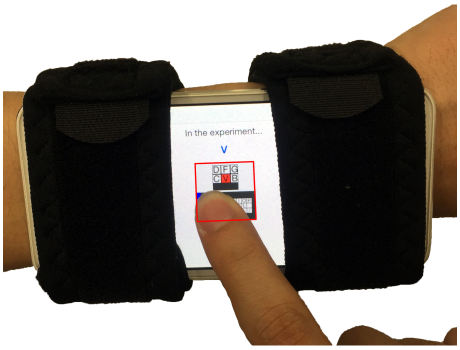

Figure 6. The smartphone attached in a landscape orientation with respect to the non-dominant hand.

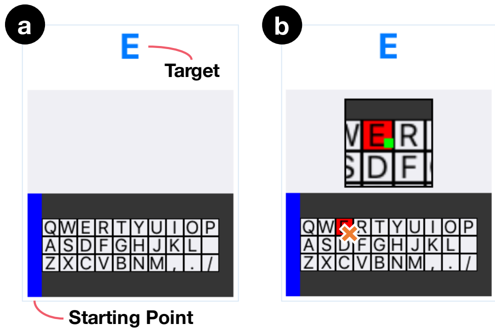


Figure 7. The application used in the experiment. a) At the start of a trial, the application displays the target ("E" in this case) and a starting point (the blue bar). b) Then, the participants select the target by touching the blue bar, dragging their finger toward the target, and selecting the target by lifting their finger.

### Participants

We recruited 16 participants (two females) aged between 21 and 25 years (M = 22.6, SD = 0.9). These 16 participants included the eight participants from our previous study [10]. All the participants had a computer science background; all the participants were right-handed; six participants used a smartwatch (period of time: 3–15 months, M = 8, SD = 5.1); all the participants were familiar with a QWERTY keyboard. Each participant received JPY 1,640 (approximately USD 15.1) after the completion of the experiment.

### Apparatus

The target selection task was implemented on an iPhone 5 smartphone (iOS 9.1, 4-inch, 1,136 × 640 pixels, 326 ppi). Similar to the approaches in (Leiva et al., 2015; Oney et al., 2013), we used the smartphone for the experiment because its touch screen is more accurate than the touch screen of existing smartwatches. A region of 18.0 × 18.0 mm (1.0 inch, 232 × 232 pixels) on the screen was used to simulate a smartwatch (similar to the approaches in Leiva et al., 2015; Oney et al., 2013); the touch events outside this region were ignored. The region was divided into two equal sub-regions: the upper region was used to display a callout; the lower region was used as the input area that displayed a tiny QWERTY keyboard. The dimension of each key was 1.6 × 1.6 mm (10 × 10 pixels), which was the same size as a key in ZShift (Leiva et al., 2015) and ZoomBoard (Oney et al., 2013). The dimension of the callout was 7.8 × 7.8 mm (50 × 50 pixels). The dimension of the region displayed by the callout was 4.4 × 4.4 mm (28 × 28 pixels); thus, the callout magnified the original image by a factor of 1.8. Similar to the approach in (Leiva et al., 2015), the smartphone was attached in a landscape orientation with respect to the non-dominant hand of the participant by using a Velcro strap (D&M Co., Ltd.; knee wrap; 842XUD2786 BLK M), as shown in Figure 6.

### Experimental Design

A repeated measures within-participant design was used in this experiment. The design had three independent variables: Presentation Method (Continuous and Discrete), Presentation Position (Fixed and Following), and Pointer Existence (NotExisting and Existing).

We used the key-entry task (not the text-entry task) as the selection task because all the participants were familiar with a QWERTY keyboard, and the selection of a letter could eliminate the effect of familiarity among participants to a greater extent than emojis or icons. Another reason for choosing the key-entry task instead of the text-entry task was that we focused on exploring the contribution of each callout design factor in helping the participants to *find* the target.

In order to examine only the effects of callout designs, we added two starting point conditions (Left or Right), as shown in Figure 7. In order to prevent participants from selecting the target without using the callout, we displayed the starting point on one side of the keyboard (left or right) as a blue bar. The participants were instructed to start a trial by touching the blue bar (*touch down*), dragging their finger toward the target, and selecting the target by lifting their finger (*touch up*). This design ensured the usage of the callout by making it necessary for the participants to touch the starting point at the beginning and forcing them to drag their finger toward the target. We presented the starting point in the left-to-right order. Therefore, for each callout design, the participants performed the task twice, i.e., once for each starting point condition.

The callout designs were presented to the participants in a random order without redundancy to counterbalance possible biases caused by the order of the conditions. For each callout design, one of the 26 targets (keys A to Z on the keyboard) was randomly presented. In summary, the experimental design involved:

2 *Presentation Method* (*Continuous* and *Discrete*) ×
2 *Presentation Position* (*Fixed* and *Following*) ×
2 *Pointer Existence* (*NotExisting* and *Existing*) ×
2 *Starting Points* (*Left* and *Right*) ×
26 *Targets*
= 416 trials per participant.

### Procedure and Task

The experiment was conducted in a calm office environment. First, the purpose of the experiment was explained to the participants. In addition, they were informed that they could abort the experiment and take a break at any time. The participants were requested to sign a consent

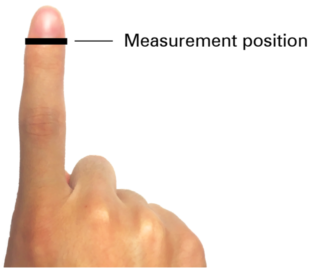


**Figure 8. Measurement position for the index finger.**

form and answer a demographics questionnaire. Then, we measured the width of the index finger of their dominant hand with a digital caliper; for the measurement, the digital caliper was aligned with the distal interphalangeal joint (Figure 8). The average width obtained was 14.9 mm (SD = 0.8), which matches the standard size for the Japanese people (Kouchi, 2012).

A smartphone was attached to the non-dominant arm of each participant. Each callout design was presented and explained to the participants through a short demonstration. Then, the participants were asked to select targets five times using each callout design as training. They were advised to use only the index finger of the dominant hand to select targets during the entire experiment. This warm-up session took an average time of approximately 3–5 minutes. Then, the actual sessions began.

In the actual sessions, a target was displayed above the region simulating the smartwatch (Figure 7). During the experiment, the participants were instructed to select the presented target as quickly and accurately as possible. After the participants selected the target, a new target was displayed. The next target was displayed immediately after participants succeeded or failed to select the current target. After each callout design was complete, the participants were requested to report their impressions regarding the selection of the targets to the experimenter and to respond to the NASA Task Load Index (NASA-TLX) questionnaires (Hart and Staveland, 1988). In this experiment, we used the Japanese version of NASA-TLX (Miyake and Kumashiro, 1993) because all the participants were Japanese. Then, the participants were requested to take a break of at least 90 seconds. After 90 seconds, the experimenter asked the participants whether they were tired. If the participants answered that they were tired, they were requested to take a break until they felt refreshed.

After all the callout designs were complete, the participants were given a questionnaire to determine their impressions about each factor. The duration of this experiment was approximately 80 minutes. The entire experiment was recorded by using screen capture in order to enable the investigation of user behavior if unusual data were obtained[1]. The comments of the participants were recorded by using a voice recorder.

[1] In the experiment, we did not use the captured data because no unusual data were obtained.

### Measurement and Analysis Methodology

The selection time was measured as the time from the *touch down* event at the starting point to the *touch up* event to select the current target. If the participants failed to select the target, we marked the selection as an error and did not include such trials in the calculation of the selection time.

## RESULTS

### Selection Speed

The selection speed for each callout design is shown in Figure 9 and Table 1. We observe that Continuous-Following-Existing condition achieved the highest selection speed; Discrete-Fixed-Existing condition showed the lowest selection speed. We analyzed the results by using a paired t-test, and the difference between these conditions was significant ($t_{15} = -4.35$, $p < 0.001$).

We analyzed the results by using a three-way repeated measure ANOVA, as shown in Figure 10. We observed a significant main effect within *Presentation Method* ($F_{1, 105} = 60.22$, $p < 0.001$); Continuous had a significantly higher selection speed. A significant interaction effect between *Presentation Method* and *Pointer Existence* ($F_{1, 105} = 4.44$, $p < 0.05$) was observed; in Continuous, Existing showed a higher selection speed; in Discrete, NotExisting achieved a higher selection speed.

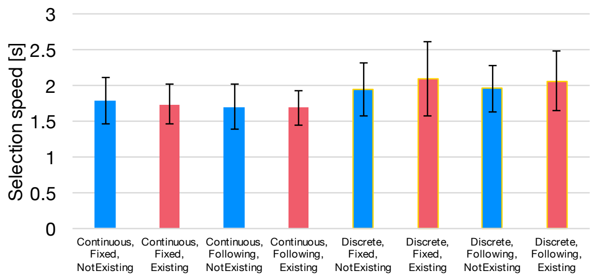


**Figure 9. The average selection speed for each condition (lower is better). Error bars indicate ± one SD.**

| Condition | Selection speed [s] | |
|---|---|---|
| Continuous, Fixed, NotExisting | 1.78 | (0.33) |
| Continuous, Fixed, Existing | 1.73 | (0.28) |
| Continuous, Following, NotExisting | 1.70 | (0.32) |
| Continuous, Following, Existing | **1.68** | **(0.24)** |
| Discrete, Fixed, NotExisting | 1.94 | (0.36) |
| Discrete, Fixed, Existing | **2.09** | **(0.52)** |
| Discrete, Following, NotExisting | 1.96 | (0.33) |
| Discrete, Following, Existing | 2.06 | (0.42) |

**Table 1. The average selection speed for each condition (lower is better). SDs are denoted in parentheses. Bold-faced type indicates the highest and the lowest selection speed.**

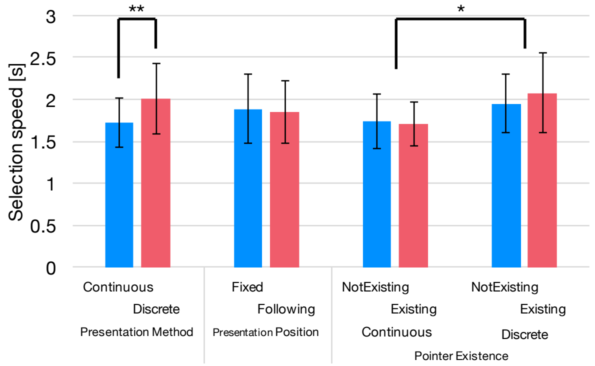


**Figure 10. The average selection speed for each design factor (lower is better). * p < 0.05, ** p < 0.001. Error bars indicate ± one SD.**

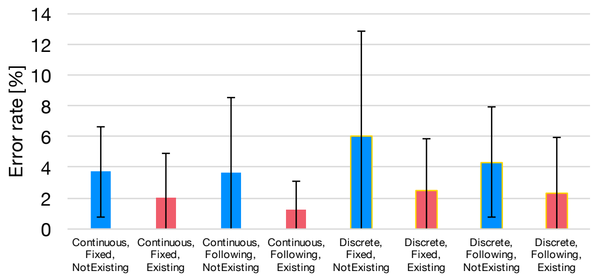


**Figure 11. The average error rate for each condition (lower is better). Error bars indicate ± one SD.**

| Condition | Error rate [%] | |
|---|---|---|
| Continuous, Fixed, NotExisting | 3.73 | (2.93) |
| Continuous, Fixed, Existing | 2.04 | (2.85) |
| Continuous, Following, NotExisting | 3.61 | (4.91) |
| Continuous, Following, Existing | **1.20** | **(1.84)** |
| Discrete, Fixed, NotExisting | **6.01** | **(6.84)** |
| Discrete, Fixed, Existing | 2.52 | (3.35) |
| Discrete, Following, NotExisting | 4.33 | (3.61) |
| Discrete, Following, Existing | 2.28 | (3.66) |

**Table 2. The average error rate for each condition (lower is better). SDs are denoted in parentheses. Bold-faced type indicates the lowest and the highest error rates.**

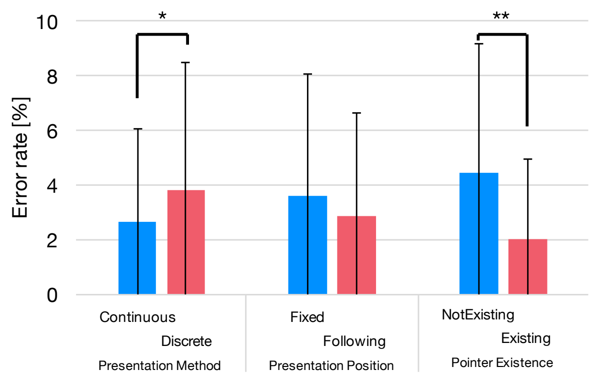


**Figure 12. The average error rate for each design factor (lower is better). * p < 0.05, ** p < 0.001. Error bars indicate ± one SD.**

## Error Rate

The error rate for each callout design is shown in Figure 11 and Table 2. We observe that Continuous-Following-Existing condition achieved the lowest error rate; Discrete-Fixed-NotExisting showed the highest error rate. We analyzed the results by using a paired t-test, and the difference between these conditions was significant ($t_{15} = 3.00$, $p < 0.01$).

We analyzed the results by using a three-way repeated measure ANOVA, as shown in Figure 12. We observed a significant main effect within *Presentation Method* ($F_{1, 105} = 4.06$, $p < 0.05$); Continuous showed a significantly lower error rate. We observed a significant main effect within *Pointer Existence* ($F_{1, 105} = 18.01$, $p < 0.001$); Existing showed a significantly lower error rate. No significant interaction effect was observed.

## Workload

The TLX score of each callout design is shown in Figure 13 and Table 3. The results show that Continuous-Following-Existing condition achieved the lowest score; Discrete-Following-Existing condition achieved the highest score. We analyzed the results by using a paired t-test, and the difference between these conditions was significant ($t_{15} = -4.32$, $p < 0.001$).

We analyzed the overall TLX scores by using a three-way repeated measure ANOVA, as shown in Figure 14. We observed a significant main effect within *Presentation Method* ($F_{1, 105} = 50.62$, $p < 0.001$); Continuous achieved a significantly lower score. A significant interaction effect between *Presentation Method* and *Pointer Existence* ($F_{1, 105} = 4.35$, $p < 0.05$) was observed; in Continuous, Existing achieved a lower score; in Discrete, NotExisting showed a lower score.

We also analyzed the TLX scale of each evaluated category by using a three-way repeated measure ANOVA. Among the six scales that TLX evaluates from different perspectives, the Mental Demand scale demonstrated a significant main effect within *Presentation Method* ($F_{1, 105} = 44.00$, $p < 0.001$); Continuous achieved a significantly lower score. The Physical Demand scale yielded a significant main effect within *Presentation Method* ($F_{1, 105} = 29.18$, $p < 0.001$); Continuous showed a significantly lower score. The Temporal Demand scale generated a significant main effect within *Presentation Method* ($F_{1, 105} = 11.54$, $p < 0.01$); Continuous achieved a significantly lower score. In the case of the Performance scale, we observed a significant main effect within *Presentation Method* ($F_{1, 105} = 16.06$, $p < 0.001$); Continuous showed a significantly lower score. We observed a significant main effect within *Pointer Existence* ($F_{1, 105} = 5.86$, $p < 0.05$); Existing achieved a significantly lower score. Further, a significant interaction effect between *Presentation Method*

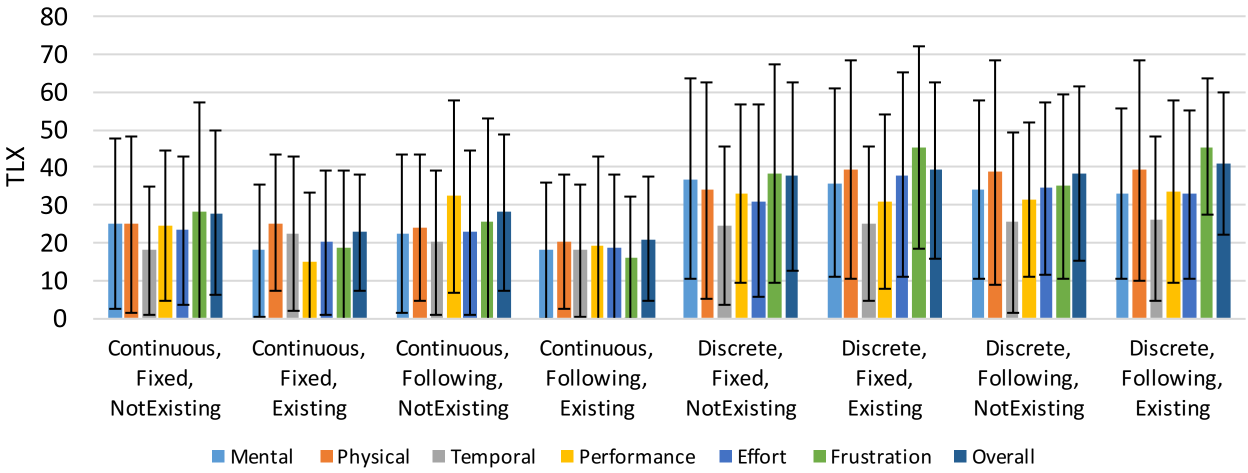


**Figure 13. TLX scales (lower is better). Error bars indicate ± one SD.**

| Condition | Overall score | |
|---|---|---|
| Continuous, Fixed, NotExisting | 27.9 | (21.7) |
| Continuous, Fixed, Existing | 22.8 | (15.5) |
| Continuous, Following, NotExisting | 28.0 | (20.8) |
| Continuous, Following, Existing | **20.9** | **(16.6)** |
| Discrete, Fixed, NotExisting | 37.6 | (25.1) |
| Discrete, Fixed, Existing | 39.2 | (23.3) |
| Discrete, Following, NotExisting | 38.2 | (23.3) |
| Discrete, Following, Existing | **40.9** | **(19.0)** |

**Table 3. The average overall TLX score for each condition (lower is better). SDs are denoted in parentheses. Bold-faced type indicates the lowest and the highest scores.**

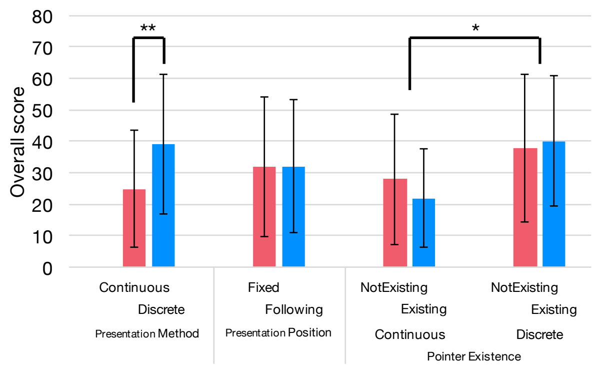


**Figure 14. The average overall TLX score for each design factor (lower is better). * p < 0.05, ** p < 0.001. Error bars indicate ± one SD.**

and *Pointer Existence* ($F_{1, 105} = 5.66$, $p < 0.05$) was observed; in Continuous, Existing showed a lower score; in Discrete, NotExisting showed a lower score. In the case of the Effort scale, we observed a significant main effect within *Presentation Method* ($F_{1, 105} = 25.73$, $p < 0.001$); Continuous yielded a significantly lower score. The Frustration scale generated a significant main effect within *Presentation Method* ($F_{1, 105} = 43.55$, $p < 0.001$); Continuous achieved a significantly lower score. Further, a significant interaction effect between *Presentation Method* and *Pointer Existence* ($F_{1, 105} = 10.44$, $p < 0.01$) was observed; in Continuous, Existing achieved a lower score; in Discrete, NotExisting showed a lower score.

### Qualitative Results

The question that we posed in the questionnaire was, "*For each of the three design factors, which level was easier to use? Continuous or Discrete, Fixed or Following, and NotExisting or Existing?*" All the participants responded that Continuous was easier to use than Discrete. Eight participants indicated that Fixed was easier to use than Following. Eight participants indicated that Following was easier to use than Fixed. Eleven participants responded that Existing was easier to use than NotExisting. These results are summarized in Table 4.

#### *Presentation Method*

The above results suggest that Continuous was easier to use than Discrete; further, Continuous shows a higher selection speed, lower error rate, and lower TLX score than Discrete. The reason for this result is that in Continuous, the finger movement of a user corresponds to the content of the callout. Comments in the questionnaire and recorded voice support this result. The participants stated that "*Continuous was more similar to the natural movement of the finger, and I was able to accomplish my task better*" (four participants) and that "*It was easier to adjust the exact position in Continuous*" (six participants).

In contrast, we supposed that Discrete could provide visual feedback of the user actions to the user because the content of a callout is changed whenever an item is selected. However, this supposition was incorrect. In the questionnaire, eight participants stated that "*In Discrete, I cannot realize fine adjustment.*"

#### *Presentation Position*

With respect to Presentation Position, we do not observe any significant difference in the performance. Further, we do not observe any significant difference in the questionnaire: eight participants answered that Fixed was easier to

| Condition | | Participants |
|---|---|---|
| Presentation Method | Continuous | 16 |
| | Discrete | 0 |
| Presentation Position | Fixed | 8 |
| | Following | 8 |
| Pointer Existence | NotExisting | 5 |
| | Existing | 11 |

**Table 4. Result of the questionnaire "*For each of the three design factors, which level was easier to use? Continuous or Discrete, Fixed or Following, and NotExisting or Existing?*"**

use than Following; in contrast, eight participants answered that Following was easier to use than Fixed. Comments in the questionnaire and recorded voice supported the above responses. Five participants stated that "*I had to follow the callout with my eyes in Following; however, I did not do so in Fixed.*" In contrast, three participants stated that "*In Following, I could recognize my action based on the change of the position of the callout*" in the questionnaire.

In addition, five participants stated that they could not perceive any difference with respect to Presentation Position. In the experiment, we used an extremely small screen. Thus, the screen size may have been too small to produce a difference between Fixed and Following.

#### *Pointer Existence*

In Existing, the pointer improves the performance because the error rate decreased. Comments in the questionnaire and recorded voice support this result. Ten participants stated that "*In Existing, selection was easier because I was able to see the actual point that I was touching" and that "It was easier to aim at the target in Existing*."

Further, 10 participants stated that "*In Discrete, NotExisting was easier to use than Existing*" in the questionnaire. The reason for this result would be that the pointer moves continuously although the content of a callout is changed discretely in response to the current selection of an item by a finger in Discrete; this behavior caused confusion, thus leading to this result.

## DISCUSSION AND FUTURE WORK

### New Insight

Our results and the results of Leiva et al. (Leiva et al., 2015) were consistent, i.e., we determined that Continuous (Presentation Method) is better. Further, we obtained a new insight that Existing (Pointer Existence) is better in Continuous. According to this result, the performance of text entry on ultra-small devices could be improved by displaying a pointer in keyboards, such as ZShift (Leiva et al., 2015), that adopt Continuous. We also observed an interesting comment in the questionnaire: two participants stated that they could not recognize the difference caused by Pointer Existence. Therefore, in future work, we will conduct additional experiments with a larger number of participants and smaller targets to examine the reason for this perception because smaller targets would impose a stronger effect of Pointer Existence.

In addition, our experiment showed that Presentation Position does not affect the selection performance. This result may be specific to ultra-small devices because it conflicts with the design of Shift (Vogel and Baudisch, 2007), which adopts Following (Presentation Position) on a device with a larger screen size rather than on an ultra-small device. Therefore, the investigation of the above hypothesis in additional controlled experiments using devices with various screen sizes is an area for future work.

### A More Practical Setup or Task for Experiments

We used a smartphone in the experiment. This design may influence the results, especially those related to the performance around the edge of the screen. Therefore, in the immediate future, we will conduct the above experiment by using a real smartwatch. We have already used the key-entry task in the experiment; therefore, we are designing another experiment to gain insights on various callout design factors in a more practical setup by investigating the performance of text entry.

### Other Design Factors

In this study, we investigated only three factors of callout design; however, various other factors – such as size, shape, and zooming ratio – could affect the selection performance.

We examined the selection performance on a single *screen size* with a single *target size*; in addition to the callout design factors, the screen size and target size could affect the selection performance. Specifically, an investigation of the selection performance for various screen sizes would reveal the effect of Presentation Position. Further, the targets in this study were clustered to the left because we used a QWERTY keyboard layout whose alphabetical keys are clustered to the left. Therefore, in future work, we will conduct an experiment using uniformly arranged targets (e.g., emoji keyboard).

These investigations would yield other guidelines, which would enable further improvements in the selection performance.

## CONCLUSIONS

We examined eight callout designs for ultra-small devices in order to extract design guidelines for callouts on such devices. From the results, we obtained the following design guidelines:

- the content of the callout should be changed continuously (the selection speed increases; the error rate and the workload decrease);
- a pointer should be displayed to indicate the position touched by the user within the callout (the error rate decreases);
- the position of the callout as a design factor does not affect the selection performance; this guideline conflicts with the design of the previous study on PDA (Vogel and Baudisch, 2007).

These guidelines will help interaction and UI designers in designing interactions and UI for ultra-small devices.

## REFERENCES


Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. In Proc. CHI 2009, ACM, 1923–1932.

Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., and To, P. Occlusion-aware menu design for digital tabletops. In Proc. CHI EA 2009, ACM, 3223–3228.

Chapuis, O., and Dragicevic, P. Effects of motor scale, visual scale, and quantization on small target acquisition difficulty. ACM Transactions on Computer-Human Interaction 18, 3 (2011), 13:1–13:32.

Chen, X. A., Grossman, T., and Fitzmaurice, G. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In Proc. UIST 2014, ACM, 615–620.

Cockburn, A., Ahlstrm, D., and Gutwin, C. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. International Journal of Human-Computer Studies 70, 3 (2012), 218–233.

Gordon, M., Ouyang, T., and Zhai, S. WatchWriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In Proc. CHI 2016, ACM, 3817–3821.

Hart, S. G., and Staveland, L. E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Human mental workload 1, 3 (1988), 139–183.

Holz, C., and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In Proc. CHI 2010, ACM, 581–590.

Hong, J., Heo, S., Isokoski, P., and Lee, G. SplitBoard: A simple split soft keyboard for wristwatch-sized touch screens. In Proc. CHI 2015, ACM, 1233–1236.

Ishii, A., Shizuki, B., and Tanaka, J. Evaluation of callout design for ultra-small touch screen devices. In Proc. CHI EA 2016, ACM, 2511–2518.

Khalilbeigi, M., Schmittat, P., Mühlhäuser, M., and Steimle, J. Occlusion-aware interaction techniques for tabletop systems. In Proc. CHI EA 2012, ACM, 2531–2536.

Komninos, A., and Dunlop, M. Text input on a smart watch. Pervasive Computing, IEEE 13, 4 (2014), 50–58.

Kouchi, M. Aist the measurement data of the hands of the japanese. https: //www.dh.aist.go.jp/database/hand/index.html (In Japanese), Last accessed on October 7, 2016, 2012.

Leiva, L. A., Sahami, A., Catala, A., Henze, N., and Schmidt, A. Text entry on tiny QWERTY soft keyboards. In Proc. CHI 2015, ACM, 669–678.

MacKenzie, I. S., Sellen, A., and Buxton, W. A. S. A comparison of input devices in element pointing and dragging tasks. In Proc. CHI 1991, ACM, 161–166.

Miyake, S., and Kumashiro, M. Subjective mental workload assessment technique - an introduction to NASA-TLX and SWAT and a proposal of simple scoring methods -. Human factors and ergonomics 29, 6 (1993), 399–408. (In Japanese).

Oney, S., Harrison, C., Ogan, A., and Wiese, J. ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In Proc. CHI 2013, ACM, 2799–2802.

Potter, R. L., Weldon, L. J., and Shneiderman, B. Improving the accuracy of touch screens: An experimental evaluation of three strategies. In Proc. CHI 1988, ACM, 27–32.

Sears, A., and Shneiderman, B. High precision touchscreens: Design strategies and comparisons with a mouse. International Journal of Man-Machine Studies 34, 4 (1991), 593–613.

Siek, K. A., Rogers, Y., and Connelly, K. H. Fat finger worries: How older and younger users physically interact with PDAs. In Proc. INTERACT 2005, Springer-Verlag, 267–280.

Vogel, D., and Balakrishnan, R. Occlusion-aware interfaces. In Proc. CHI 2010, ACM, 263–272.

Vogel, D., and Baudisch, P. Shift: A technique for operating pen-based interfaces using touch. In Proc. CHI 2007, ACM, 657–666.

Xia, H., Grossman, T., and Fitzmaurice, G. NanoStylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In Proc. UIST 2015, ACM, 447–456.