Pressure-Based One-Handed Interaction Technique for Large Smartphones Using Cursor

Kyohei Hakka
University of Tsukuba
Tsukuba, Ibaraki, Japan
hakka@iplab.cs.tsukuba.ac.jp

Toshiyuki Ando
University of Tsukuba
Tsukuba, Ibaraki, Japan
ando@iplab.cs.tsukuba.ac.jp

Buntarou Shizuki
University of Tsukuba
Tsukuba, Ibaraki, Japan
shizuki@iplab.cs.tsukuba.ac.jp

Shin Takahashi
University of Tsukuba
Tsukuba, Ibaraki, Japan
shin@iplab.cs.tsukuba.ac.jp

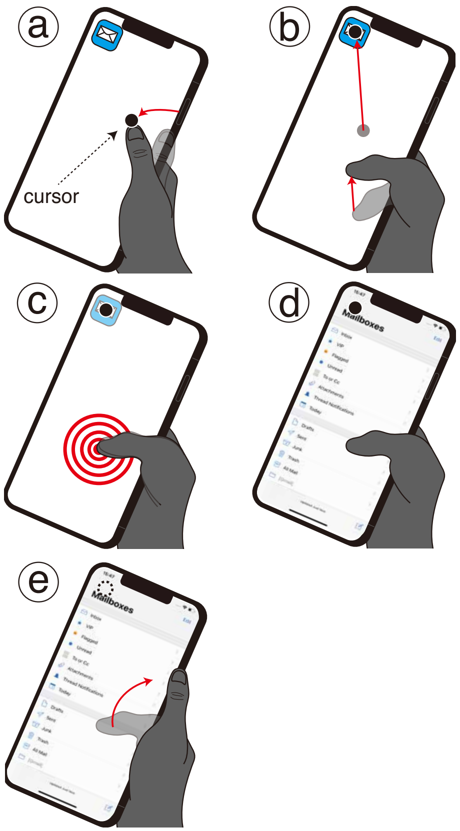

Figure 1: The usage of our technique. (a) The cursor appears at the center of the touch area when the user performs a bezel-swipe. (b) The cursor moves when the user drags the finger. (c, d) The touch-down/up event is generated when the user increases/decreases the touch pressure. (e) The cursor disappears when the user releases the finger.

ABSTRACT

We show a one-handed interaction technique using a cursor for large smartphones. When applying our technique, a user can generate a touch-down event at the position of the cursor by increasing the touch pressure, and a touch-up event by decreasing the touch pressure. This pressure-based approach enables the user to perform various single-touch gestures (e.g., tap, swipe, drag, and double-tap) at the position of the cursor. The results of our pilot study, for which we employed a prototype that uses a bezel-swipe as a trigger to show the cursor, are as follows: 1) it took 1.7 seconds for the user to perform a single-touch gesture using the cursor, 2) the success rate was 92.71%, and 3) the user could access the entire screen with a stable grip.

Asian CHI Symposium '19, May 06, 2019, Glasgow, UK

KEYWORDS

touch pressure, single-touch gesture, touch screen, bezel-swipe

INTRODUCTION

It is difficult for a user to interact with a large smartphone with one hand without changing a grip of the smartphone because the reach of the thumb is limited [1?], although a user prefers to interact with a smartphone with one hand [11, 18]. Moreover, the user's grip becomes unstable if it is changed during one-handed interaction. This may cause the user's finger or palm to touch the screen unintentionally, and thus cause unintended interactions; furthermore, the user may drop the smartphone.

To address this problem, we are exploring a one-handed interaction technique using a cursor. Several similar techniques to ours, using a cursor, have been proposed (e.g., BezelCursor [16] and Extendible Cursor [12]). However, these were proposed as techniques for selecting (or tapping) unreachable targets. Therefore, the user cannot use these techniques to input single-touch gestures, which are gestures performed with only one finger, such as a tap, drag, double-tap, or swipe, using the cursor, except for a tap. On the other hand, our technique enables the user to input single-touch gestures. Specifically, our technique uses the touch pressure to enable the user to operate the cursor in a similar manner to a computer mouse or touchpad: a touch-down event is generated at the position of the cursor when the user increases the touch pressure; a touch-up event is generated at the position of the cursor when the user decreases the touch pressure.

In this paper, we show a prototype of our technique that works on iOS (Figure 1). This prototype uses a bezel-swipe [20] as a trigger. In our prototype, the user 1) performs a bezel-swipe to trigger the cursor operation mode (Figure 1a), 2) moves the cursor by dragging the finger on the screen (Figure 1b), 3) generates the touch-down event at the position of the cursor by increasing the touch pressure (Figure 1c), 4) generates the touch-up event by decreasing the touch pressure (Figure 1d), and 5) releases the finger from the screen to end the cursor operation mode (Figure 1e). We also report the results of a pilot study, in which we investigated the performance of this technique based on our prototype.

RELATED WORK

Many techniques have been proposed for one-handed interactions with large smartphones. Four approaches have been adopted with respect to these previous techniques: using a cursor, moving the screen contents, reducing the screen contents to a small proxy area, and transferring touch events to another position on the screen.

Using a Cursor

Kim et al. [12] proposed Extendible Cursor, which uses a bezel-swipe or a large area touch by the pad of the thumb as a trigger. Furthermore, Kim et al. showed that a bezel-swipe is superior to a large area touch as a trigger, and that the user can perform a bezel-swipe regardless of the orientation of the smartphone. Li et al. [16] also proposed a cursor technique using a bezel-swipe as the trigger. CornerSpace [24] generates four corner buttons, which denote each corner of the screen, when the user performs a bezel-swipe; by selecting a corner button, the user can show a cursor at the corresponding corner of the screen. Chang et al. proposed TiltCursor [2], which is a technique that uses a tilting gesture as the trigger. When the user tilts the smartphone by 35 degrees or more, TiltCursor triggers the cursor operation mode. In these techniques, a tap event is generated at the position of the cursor when the user releases the finger during cursor operation mode. However, there are two main problems with this design. First, the user cannot input a single-touch gesture, with the exception of a tap. Second, the user cannot continuously select objects using the cursor, because the cursor operation mode ends when the user releases the finger.

While our technique uses a cursor, it differs from the previous techniques in that a touch event is generated based on the touch pressure. With this design, our technique addresses the problems of the previous cursor techniques.

Moving Screen Contents

Techniques that allow the user to move the screen contents to the thumb based on the distance that a finger is dragged have been proposed [2, 4, 12, 14, 22] using various triggers. Sliding Screen [12] and MovingScreen [22] use a bezel-swipe. TiltSlide [2] uses a tilting gesture. IndexAccess [4] and the technique proposed by Le et al. [14] requires the user to drag the forefinger along the back of the smartphone. On the other hand, [5, 15] are techniques for moving the contents to the lower half of the screen. As the trigger, Apple's Reachability [5] uses a swipe down on the bottom edge of the screen on bezel-less models and a double-tap on the home button on other models, while PalmTouch [15] uses the gesture of placing the palm on the screen.

In these techniques, because the screen contents are moved, a part of the content is extruded from the screen; thus, these techniques do not support long drags; for example, the drag from the top to the bottom of the screen is still difficult.

Reducing Screen Contents

TiltReduction [2] reduces the screen contents to a small proxy area, which is displayed within the reachable area of each user's thumb when they tilt the smartphone. Similarly, one-handed operation mode [10] of the Samsung Galaxy reduces the screen contents to the proxy area whilst the user performs a swipe from a bezel to the center, and then swipes back to the bezel. When applying these

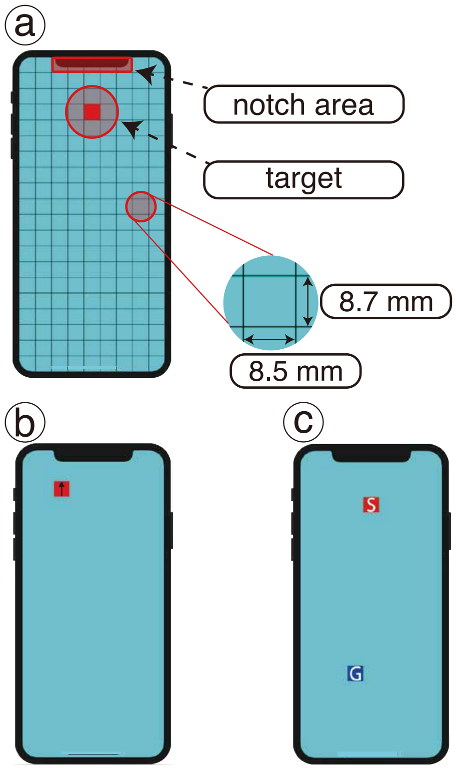

Figure 2: The screen during the experiment (a: tap and double-tap task, b: swipe task, c: drag task). The ruled lines were not displayed on the actual screen; the cells were laid out without gaps.

techniques, it is necessary to increase the reduction rate as the screen becomes larger. However, this may cause Fat finger problem [21] and occlusion because the objects in the proxy area become small.

Transferring Touch Event

In ExtendedThumb [13], the user 1) triggers a pointer operation mode by a double-tap; 2) moves the pointer to the target position by dragging the finger; 3) releases the finger to determine the position of the pointer; and 4) inputs the touch gesture that the user wants to transfer to the position of the pointer. Unlike this technique, our technique is capable of continuous interaction with unreachable areas. Gaze and Touch [19] is a technique proposed for tablets; it detects the user's gaze and transfers the touch event to the gaze position. In contrast, we only use touch information.

SYSTEM DESIGN

We implemented a prototype using iPhone's ForceTouch [8], from which we obtained the touch pressure. In the prototype, the user triggers the cursor operation mode by a bezel-swipe, and the cursor then appears at the center of the touch area. The cursor operation mode continues until the user releases the finger from the screen. In the cursor operation mode, when the user drags the finger, the cursor moves in the same direction as the finger. The distance moved by the cursor is calculated as the product of the moving distance of the finger and the Control-Display ratio (CD ratio); Although the CD ratio should be defined for each user, we defined it as 3.0 in the prototype, as we felt that this ratio would be easy to use. The cursor is a black circle with a white edge and has a radius of 1.5 mm.

The user can generate a touch-down event at the position of the cursor by increasing the touch pressure to or beyond a pre-defined threshold; a touch-up event is generated at the position of the cursor when the user decreases the touch pressure below the threshold. Our technique also provides haptic feedback by vibrating the smartphone when a touch-down event is generated. In the prototype, we defined the threshold as 2.0. According to Apple's API documentation [9], the iPhone's force sensor API delivers unitless force values. Values of approximately 1.0 represent ordinary touch pressure; the higher the touch pressure, the higher the value. Although Apple does not state how these values are converted into Newtons (N), Nelson [17] showed that an iPhone force sensor value of 0-6.67 corresponds to 0-4 N. Accordingly, our threshold corresponds to 1.2 N.

PILOT STUDY

We carried out a pilot study to investigate the performance of our prototype.

Setup

The first author (22 years old) was the participant and we used an iPhone XS Max (157.5 mm $\times$ 77.4 mm$\times$7.7 mm; 6.5 inches) as the smartphone in this study. The screen was divided into 9×20 ($=$

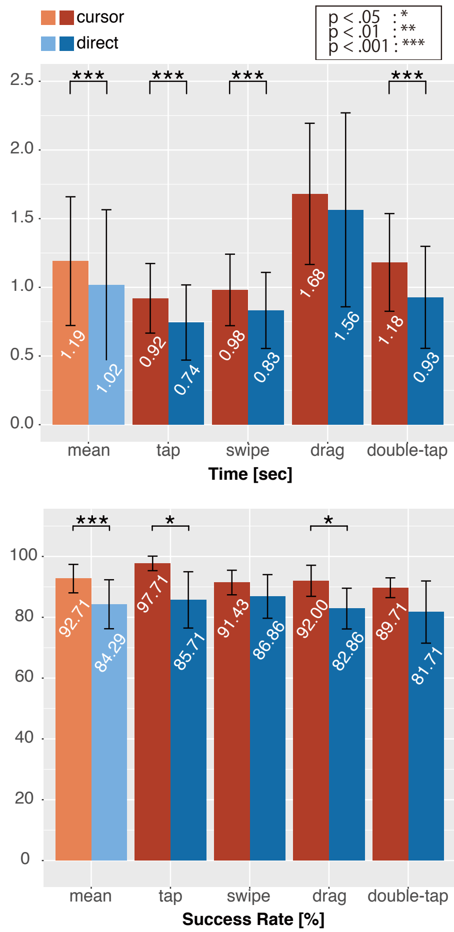

Figure 3: The time and success rate of the pilot study. The error bars indicate the standard deviation.

180) cells. The cells had dimensions 8.5 mm $\times$ 8.7 mm (Figure 2), which we defined to make them as similar as possible to Apple's recommended minimum target size (8.4 mm $\times$ 8.4 mm) [7].

Method

The participant sat on a chair and held the smartphone in the right hand (dominant hand). Then, the participant performed four tasks: *tap task*, *swipe task*, *double-tap task*, and *drag task*. In the *tap/double-tap tasks*, the participant tapped/double-tapped the target (Figure 2a). In the *swipe task*, the participant performed a swipe on the target following the arrow (Figure 2b). In the *drag task*, the participant dragged the target marked "S" to the target marked "G" (Figure 2c).

The participant performed each task using two techniques. The first was our technique (*cursor*), where the participant input all of the gestures using the cursor. The other technique was direct touch (*direct*): the participant input all of the gestures by touching the screen directly. The participant first performed all four tasks with *cursor*, and then *direct*. For each technique, the participant completed the tasks in the following order: *tap task*, *swipe task*, *drag task*, and *double-tap task*.

A trial was to correctly input each gesture once; if the gesture was correct, the target was updated; otherwise, the participant input the gesture to the target again. The participant performed 35 trials in a session and completed five sessions for each task and each technique. All of the cells were used once as the target, and were selected in a random order, except for in the *drag task*. In the case of the *drag task*, all of the cells became the "S" target once, in a random order. In total, we collected data from 1400 (1 participant $\times$ 35 trials $\times$ 5 sessions $\times$ 4 tasks $\times$ 2 techniques) trials.

Results

We analyzed the time per trial and success rate per session, which are shown in Figure 3. We compared the average values on all tasks using Welch's t-test. The tests show that *direct* was significantly faster than *cursor*, while the success rate of *cursor* was significantly higher than *direct*. To evaluate the stability of the smartphone, we differentiated the values obtained from the built-in acceleration sensor and the gyro sensor (Figure 4). We compared them using Welch's t-test; both values of *cursor* were significantly lower than *direct* for all tasks.

Figure 5 shows the gesture footprints of all tasks, which were much smaller in *cursor*. We calculated the effective area [3] (blue rectangles in Figure 5). The blue rectangle indicates the area that the participant requires to operate the entire screen using our technique, which occupied only 14% of the screen.

DISCUSSION

We anticipated the longer time and higher success rate seen with use of the cursor (Figure 3); a user can perform gestures roughly but quickly with touch operations and slowly but accurately with

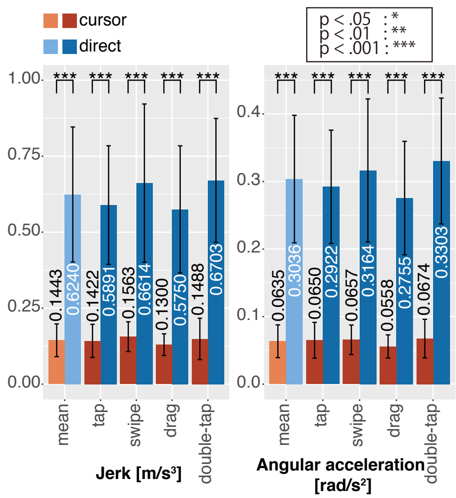

Figure 4: Jerk and angular acceleration of the smartphone in the pilot study. The error bars indicate the standard deviation.

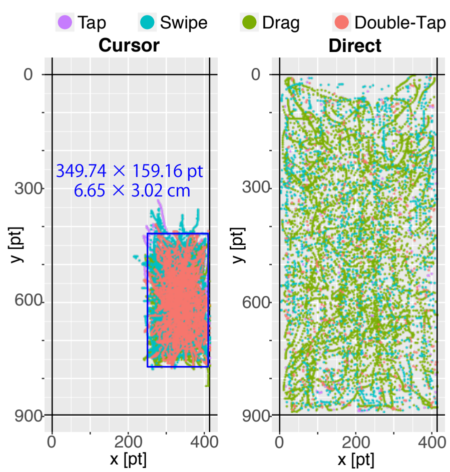

Figure 5: Gesture footprints in the pilot study. The black rectangle represents the smartphone screen; the blue rectangle represents the effective area.

the cursor because the cursor solves Fat finger and occlusion problems. Moreover, this difference was small, and all of the operations were performed using the cursor in the experiment of *cursor*; therefore, the difference would decrease if touch operations were used for reachable targets.

The jerk and angular acceleration of *cursor* were significantly lower in all tasks, and only 14% of the screen was required to operate the cursor, as shown in Figure 5. These results suggest that, using our technique, the user can access the entire screen in a stable grip without extending the thumb. This shows that our technique is promising for one-handed interactions.

In our pilot study, the CD ratio and threshold of the touch pressure were pre-defined constants. However, these values may affect the performance of our technique. Therefore, it is necessary to carry out an additional study using various CD ratios and touch pressure thresholds to investigate the cursor performance in more detail.

Wilson et al. [23] reported that, based on visual feedback, users can control 10 levels of touch pressure with their fingers on a handheld device with up to 85% accuracy. In contrast, with our technique, we only identify two levels (high and low) of touch pressure. Therefore, it would be possible to add various functions to the cursor by increasing the level of touch pressure to be identified. Examples include the features provided by iPhone's 3DTouch [6], which can display a shortcut menu and a preview, or the zooming functionality of photos and maps; zooming is difficult for users during one-handed interactions because it is usually performed with a pinch gesture.

CONCLUSION

We showed a pressure-based one-handed interaction technique using a cursor. In our technique, in the cursor operation mode, the user can generate a touch-down event at the position of the cursor by increasing the touch pressure, then generate a touch-up event by decreasing the touch pressure. This pressure-based approach enables the user to perform a single-touch gesture (e.g., a tap, swipe, drag, and double-tap) at the position of the cursor. The results of the pilot study of our prototype indicated that the average time required to perform a single-touch gesture using the cursor was 1.19 sec, and the success rate was 92.71%. Moreover, the variations in the values of the acceleration sensor and the gyro sensor were much smaller when our technique was used rather than direct touch. This means that the user can access the entire smartphone screen in a stable grip using our technique. Furthermore, the user only used 14% of the screen to operate the cursor.

FUTURE IMPACT

In addition to supporting one-handed interactions with large smartphones, our technique allows users to operate tablets with one hand. Furthermore, the technique would also allow users to operate smartwatches using a smartphone as a touchpad, thus solving Fat finger and occlusion problems because the user can operate the smartwatch with the cursor.

REFERENCES

[1] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. 2014. Modeling the Functional Area of the Thumb on Mobile Touchscreen Surfaces. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1991–2000. https://doi.org/10.1145/2556288.2557354

[2] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1499–1508. https://doi.org/10.1145/2702123.2702425

[3] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. 2018. Use the Force Picker, Luke: Space-Efficient Value Input on Force-Sensitive Mobile Touchscreens. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 661, 12 pages. https://doi.org/10.1145/3173574.3174235

[4] Shiori Hidaka, Tetsuaki Baba, and Paul Haimes. 2016. IndexAccess: A GUI Movement System by Back-of-Device Interaction for One-Handed Operation on a Large Screen Smartphone. *International Journal of Asia Digital Art and Design Association* 20, 2 (2016), 41–47. https://doi.org/10.20668/adada.20.2_41 Asia Digital Art and Design Association.

[5] Apple Inc. 2014. Reachability - iPhone User Guide. https://help.apple.com/iphone/11/?lang=en#/iph66e10a71c (accessed on 12 February 2019).

[6] Apple Inc. 2018. 3D Touch - iOS - Apple Developer. https://developer.apple.com/ios/3d-touch/ (accessed on 12 February 2019).

[7] Apple Inc. 2018. Adaptivity and Layout - Visual Design - iOS - Human Interface Guidelines - Apple Developer. https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/ (accessed on 12 February 2019).

[8] Apple Inc. 2018. Force Touch - Apple Developer. https://developer.apple.com/macos/force-touch/ (accessed on 12 February 2019).

[9] Apple Inc. 2018. UITouch-UIKit | Apple Developer Documentation. https://developer.apple.com/documentation/uikit/uitouch (accessed on 12 February 2019).

[10] Samsung Inc. 2018. How do I use the reduce screen size of one-handed operation on Note4 Samsung Support HK_EN. https://www.samsung.com/hk_en/support/mobile-devices/how-do-i-use-the-reduce-screen-size-of-one-handed-operation-on-note4/ (accessed on 12 February 2019).

[11] Amy K. Karlson and Benjamin B. Bederson. 2006. *Studies in One-Handed Mobile Design: Habit, Desire and Agility*. Technical Report. Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4ALL '98).

[12] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. 2012. Interaction Techniques for Unreachable Objects on the Touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference (OzCHI '12)*. ACM, New York, NY, USA, 295–298. https://doi.org/10.1145/2414536.2414585

[13] Jianwei Lai and Dongsong Zhang. 2015. ExtendedThumb: A Target Acquisition Approach for One-Handed Interaction With Touch-Screen Mobile Phones. *IEEE Transactions on Human-Machine Systems* 45, 3 (2015), 362–370. https://doi.org/10.1109/THMS.2014.2377205 IEEE.

[14] Huy Viet Le, Patrick Bader, Thomas Kosch, and Niels Henze. 2016. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 27, 10 pages. https://doi.org/10.1145/2971485.2971562

[15] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm As an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 360, 13 pages. https://doi.org/10.1145/3173574.3173934

[16] Andy Li, Hongbo Fu, and Zhu Kening. 2016. BezelCursor: Bezel-Initiated Cursor for One-Handed Target Acquisition on Mobile Touch Screens. *International Journal of Mobile Human Computer Interaction* 8 (2016), 1–22. IGI Global.

[17] R. Kevin Nelson. 2015. Exploring Apple's 3D Touch. https://medium.com/@rknla/exploring-apple-s-3d-touch-f5980ef45af5 (accessed on 12 February 2019).

[18] Alexander Ng, Stephen A. Brewster, and John H. Williamson. 2014. Investigating the Effects of Encumbrance on One- and Two- Handed Interactions with Mobile Devices. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1981–1990. https://doi.org/10.1145/2556288.2557312

[19] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 301–311. https://doi.org/10.1145/2984511.2984514

[20] Volker Roth and Thea Turner. 2009. Bezel Swipe: Conflict-Free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In *Proceedings of the 27th Annual ACM Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1523–1526. https://doi.org/10.1145/1518701.1518933

[21] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 9th IFIP TC 13 International Conference on Human-computer Interaction (INTERACT' 05)*. Springer-Verlag, Berlin, Heidelberg, 267–280. https://doi.org/10.1007/11555261_24

[22] Hsin-Ruey Tsai, Da-Yuan Huang, Chen-Hsin Hsieh, Lee-Ting Huang, and Yi-Ping Hung. 2016. MovingScreen: Selecting Hard-To-Reach Targets with Automatic Comfort Zone Calibration on Mobile Devices. In *Proceedings of the 18th Interactional Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 651–658. https://doi.org/10.1145/2957265.2961835

[23] Graham Wilson, Craig Stewart, and Stephen A. Brewster. 2010. Pressure-based Menu Selection for Mobile Devices. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 181–190. https://doi.org/10.1145/1851600.1851631

[24] Neng-Hao Yu, Da-Yuan Huang, Jia-Jyun Hsu, and Yi-Ping Hung. 2013. Rapid Selection of Hard-To-Access Targets by Thumb on Mobile Touch-screens. In *Proceedings of the 15th Interactional Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 400–403. https://doi.org/10.1145/2493190.2493202

SHORT BIOS OF AUTHORS

Kyohei Hakka

He is currently an undergraduate student of the College of Information Science, University of Tsukuba. He graduated from National Institute of Technology, Ishikawa College, Japan. He is interested in the field of human computer interaction.

Toshiyuki Ando

He is currently a graduate student of the Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba. He graduated from the College of Media Arts, Science and Technology, School of Informatics, University of Tsukuba, Japan. He is interested in the field of human computer interaction.