Pressure-sensitive Zooming-out Interfaces for One-handed Mobile Interaction

Kenji Suzuki
Yahoo Japan Corporation
Tokyo Garden Terrace Kioicho
1-2 Kioicho, Chiyoda-ku,
Tokyo 102-8282 Japan
kensuzuk@yahoo-corp.jp

Ryuuki Sakamoto
Denqvision Inc.
ONZE212, 11-17 Sakae-cho, Nerima-ku, Tokyo
1760-006 Japan
rkskmt@denqvision.com

Daisuke Sakamoto and Tetsuo Ono
Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo,
Hokkaido 060-0814 Japan
sakamoto@ist.hokudai.ac.jp
tono@ist.hokudai.ac.jp

ABSTRACT
We present new alternative interfaces for zooming out on a mobile device: Bounce Back and Force Zoom. These interfaces are designed to be used with a single hand. They use a pressure-sensitive multitouch technology in which the pressure itself is used to zoom. Bounce Back senses the intensity of pressure while the user is pressing down on the display. When the user releases his or her finger, the view is bounced back to zoom out. Force Zoom also senses the intensity of pressure, and the zoom level is associated with this intensity. When the user presses down on the display, the view is scaled back according to the intensity of the pressure. We conducted a user study to investigate the efficiency and usability of our interfaces by comparing with previous pressure-sensitive zooming interface and Google Maps zooming interface as a baseline. Results showed that Bounce Back and Force Zoom was evaluated as significantly superior to that of previous research; number of operations was significantly lower than default mobile Google Maps interface and previous research.

Author Keywords
Zooming user interface; ZUI; mobile interaction; one-handed interaction; user study.

ACM Classification Keywords
H.5.2. User Interfaces: Interaction styles (e.g., commands, menus, forms, direct manipulation).

INTRODUCTION
Zooming is one of the major operations performed on mobile devices. People zoom frequently while browsing maps, photos and videos, and documents. Various user interfaces for zooming have been presented in the field of human–computer interaction (HCI); however, single-hand zooming interface on mobile devices have not been well explored. The most common zooming operation on mobile devices is pinching. Users look at, for example, a map by opening and closing two fingers on a touch display. This technique requires two-handed operation; those who use a single hand run the risk of dropping their mobile device. Another well-known zooming interface is the double tap, in which users quickly tap on the display twice to zoom in and scale up content. However, this method only allows users to zoom in, and the zoom level is predefined; users cannot freely control the zoom level and need to use other commands for custom zooming. The mobile Google Maps zooming interface supports single-handed zooming interaction; user can zooming-in and -out by pressing their finger on the screen and sliding their finger up and down for zooming-in and out. We compare our interfaces with mobile Google Maps zooming interface as a baseline.

MobileHCI '18, September 3–6, 2018, Barcelona, Spain

ACM ISBN 978-1-4503-5898-9/18/09 $15.00
https://doi.org/10.1145/3229434.3229446

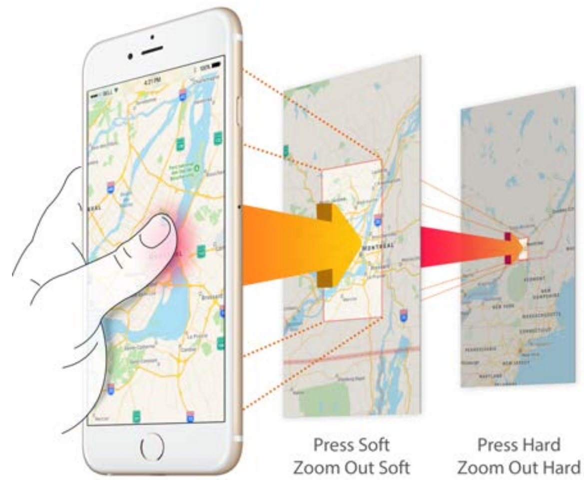

Figure 1: Force Zoom technique

We present new alternative interfaces for zooming out on a mobile device. The key to the interfaces is the use of Force Touch, a pressure-sensitive multitouch technology where the display senses the pressure with which the user is pressing down on it. We designed two user interfaces for zooming out: Bounce Back and Force Zoom. Bounce Back senses the intensity of pressure while the user is pressing down on the display. When the user releases his or her finger from the display, the device zooms out according to the intensity of the pressure. Force Zoom also senses the intensity of pressure, but immediately zooms out when the user presses down on the display. The user can adjust the zoom level by changing the intensity of the pressure (Figure 1). We investigated the efficiency and usability of the interfaces by comparing them to a current standard zooming interface and a previously presented pressure-sensitive zooming interface [11]. The major contributions of this work include the following:

- Presenting and demonstrating pressure-sensitive zooming-out manipulation interfaces on mobile devices. The interfaces are designed to support one-handed interaction on mobile devices.
- Performing a user study to compare four one-handed mobile interaction methods; Bounce Back, Force Zoom, default mobile Google Maps interface, and previous research, GraspZoom [11].
- Presenting the results of our usability study, which illustrate how the zooming-out interfaces contribute to one-handed mobile interaction and how easily they can be integrated into current user interfaces on mobile devices.

RELATED WORK

User interfaces for pressure-sensing mobile touch screens have been well studied in the field of HCI [16]. Miyaki and Rekimoto presented a zooming user interface, GraspZoom, that involved grasping the mobile device [11]. Although their technique is similar to ours, their work focused on hardware and software construction and not on interaction design; they have generally not performed usability studies. Other techniques have utilized force as well. For example, ForceTap utilized Z-axis accelerations to sense tapping intensity [5], and Force Gesture utilized tangential forces on the screen of the mobile device [6]. Harrison presented a shear input technique for touchscreens, which also utilized the tangential forces [7]. It demonstrated zooming manipulation by sharing up and down the screen. Clarkson et al. investigated the possibility of using pressure as input for mobile devices [3]. Stewart et al. conducted a series of studies to understand the characteristics of pressure-based input for mobile devices [18]. Recently, touch force sensing by a built-in barometer was presented [19]. All in all, we consider the pressure-based interface to be one of the major current research trends in mobile HCI, and our work to be the first study to confirm the usability of a pressure-sensitive zooming interface on a mobile device. Our results add to previous work.

However, we are interested in supporting one-handed interaction on mobile devices [4]. Our target manipulation is zooming and navigation. TapTap and MagStick were designed to enable users to point to a target using a single hand [17]. Rekimoto demonstrated a tilt-based interaction method for navigation [15]. Sensor synaesthesia demonstrated another zooming technique by tilting a mobile device [14]. Holman presented an auxiliary finger input interface by adding a touch-sensitive sensor on a side of mobile device [8]. A semiautomatic approach for zooming and navigation can be an option as an intelligent user interface [9]. Understanding the use of one-handed mobile interaction is an active area of mobile HCI research [12] [13]. In this work, we investigate the use of a pressure-based interface for one-handed interaction. Our results and insight contribute to this area of mobile HCI research.

Figure 2: Bounce Back (top) and Force Zoom (bottom)

BOUNCE BACK AND FORCE ZOOM

The key behind our alternative zooming user interfaces is to use the intensity of pressure on the touch display to zoom out. Here we briefly introduce the two techniques. Note that we only focus on zooming out; however, users also need to be able to zoom in to browse media and maps. We expect these interfaces to be used with current standard zooming-in interfaces, such as the double tap. We can integrate Bounce Back and Force Zoom with standard zooming user interfaces on mobile devices.

Bounce Back

The design of Bounce Back is inspired by a spring. When a spring is compressed from its resting position, it exerts an opposing force approximately proportional to its change in length. If a spring on a table is compressed by a user's finger, and then the user releases his or her finger from the spring, the spring immediately and quickly jumps up from the table. This is the metaphor behind the zooming-out manipulation. The user presses down on the display of the mobile device, perhaps with the thumb, and then releases the thumb. The view zooms out just as the spring jumps back from the table (Figure 2, top). The intensity of the pressure is reflected in the elevation of the jump, which in this case is the zoom level. The intensity of the pressure is made visible as a shadow under the user's finger on the display, so the user will know how far the device will zoom out. The user is able to adjust the intensity by looking at the shadow under his or her finger.

Force Zoom Technique

We can also use the intensity of pressure as the zoom level itself; for example, strong pressure is associated with the zooming-in manipulation. Various zooming-in manipulations (e.g., the double tap and long tap) are available on mobile devices, and these are considered standards. Thus, we modify the interface design such that strong pressure is used to zoom out (Figure 2, bottom). Consider a situation in which a larger map is behind the current map level (Figure 1). The intensity of pressure is used to adjust the zoom level. When the user presses down on the display quickly, the zoom level changes quickly, and when the user presses the display slowly, the device zooms slowly. When the user releases his or her finger, zooming stops.

USER STUDY

We compared our Bounce Back and Force Zoom techniques with the previous work (GraspZoom [11]) and the default mobile Google Maps zooming user interface (Default ZUI). The task was multiscale target selection, in which participants were asked to select a target on the screen using the four interfaces. We measured performance, collected usability feedback from the participants, and analyzed the collected data.

Equipment

All user interfaces were implemented on an iPhone 6 with iOS 10.3.3. The display was a 4.7-in screen with 1,334 × 750 resolution. Participants were instructed to hold the smartphone in their dominant hand and to place their nondominant hand on the table, or in their lap, and never use it to control the mobile device.

Task: Multiscale Target Selection

We selected a multiscale target selection task [10]; that is, the participants were asked to reach and validate a series of 10 square targets with sides 120 px, each of which was 10,176 px away from the center of the starting point. This distance was obtained using the following formula:

$$\left(\frac{1(default\_zooming\_level)}{0.15(target\_zooming\_level)}\right)\sqrt{800^2+1300^2}$$

where 800 (px) is the width (x-axis) and 1,300 (px) is the height (y-axis). The default zooming level was 1 and target zooming level was 0.15, so this value was the z-axis. We used continuous zooming, so there was no explicit zoom level.

Session: Zooming out and Zooming out/in

We prepared two sessions to compare the usability and performance of interfaces; 1) Zooming out session and 2) Zooming out/in session. First, task of the zooming out session mainly consists of zooming out operations. At the beginning of a task, the zoom level was set at maximum and the target was not visible. The user first needed to find the target by zooming out using the four manipulation techniques. Generally speaking, the target was placed far enough away that the user needed to zoom out fully. Validating the target required capturing it in a box with a 40 px margin area, so participants were required to zoom in sufficiently on the target. Once the participants validated a target, the target disappeared, and a new task began; the view was set back to the starting position. Another target instantly appeared at another position. This task mimicked what users have to do when searching for a location on a map (zooming in and out).

Next, in the zooming out/in session, the user asked to zoom out first and then needs to zoom in fully in the interface. Likewise, at the beginning of a task, the zoom level was set at maximum and the target was not visible. First, the user need to fully zoom out using four manipulation technique to find a target, and then the target is appeared on far enough away that the user needed to zoom in fully. The user zooms in to validate the target with zooming in manipulation technique. Double tap was activated in Bounce Back, Force Force Zoom, and Default ZUI. Double tap was disabled in GraspZoom. “DoubleTap and Slide” technique (Default Google Map interface) was activated in Default ZUI. Force-based zooming was activated in GraspZoom. Once the participants validated a target, the target disappeared, and a new task began; the view was set back to the starting position. The user starts to zoom out to find next target.

Condition: User Interface

The condition in this study was the user interface: Bounce Back, Force Zoom, GraspZoom [11], and Default ZUI as a baseline condition. We did not include a button-based user interface as a baseline. Button-based interfaces, such as pressing + to zoom in and – to zoom out, are not supported in major map-browsing applications on mobile devices. For example, button-based interfaces disappeared from the Google Maps application for Android OS in April 2012 (ver. 6.7). Likewise, iOS Maps has not supported the + and – button interface since 2010 (around the time of iOS 4). Thus, we considered the use of button-based interfaces on mobile devices to be out of date and did not include one as a baseline condition in this study.

Participants and Procedure

Twenty-four volunteers (16 females ranging in age from 22 to 35 years old; mean age = 28.88 years; SD=4.3) participated in this study. All participants owned a smartphone: seven owned an Android (29.2%), twenty-three owned an iPhone (95.8%), and one owned iPad (25.0%). All smartphone users had previous experience with zooming operation on a smartphone, and six had previous experience with Default ZUI of this study. In other word, other eighteen participants did not know the Default ZUI (mobile Google Maps interface). Zooming operation is often used for zooming in maps, photo/video, internet browser. Pinching gesture is best-known technique for zooming. Generally, double tap technique is used for zooming in; however, zooming-out operation is not frequently used with one-handed manipulation method.

Participants were asked to use the four different zooming methods (Bounce Back, Force Zoom, GraspZoom [11], and Default ZUI) 10 times each. The method of presentation was fully counterbalanced. All trials included 10 target selection tasks, and each participant completed all tasks. Zooming out session was always first and Zooming out/in session was next. Ultimately, participants completed 80 target selection tasks using the four methods. The target to be selected was preprogrammed (i.e., it was not randomly presented) to keep experimental conditions consistent among participants. Prior to the experiment, the participants completed a trial session with each method to become familiar with interface.

Implementation and Interaction

Here we briefly explain the actual implementation of the interfaces. Note that the pinching gesture was disabled in all user interfaces; we focus on the single-handed interaction in this study. However, double tapping (zooming in) was implemented in all methods, except the GraspZoom method. Likewise, swiping (navigation) was implemented to complete the task.

Bounce Back and Force Zoom

Here we briefly review differences from Default ZUI. Bounce Back and Force Zoom did not support any of the methods that were supported by Default ZUI, except for the use of the double tap to zoom in.

GraspZoom (Miyaki and Rekimoto, MobileHCI '09 [11])

This method is a replication of GraspZoom interface [11]. We use a pressure sensor on iPhone for the implementation of GraspZoom, instead of using hardware sensor that was used in original implementation of GraspZoom. In this method, we used the intensity of pressure as the zoom level itself; for example, strong pressure is associated with the zooming-in or -out manipulation. Zooming method is same as Force Zoom, but the user can switch the zooming mode by sliding their finger on the screen. All zooming-in and -out manipulation can be done by using GraspZoom.

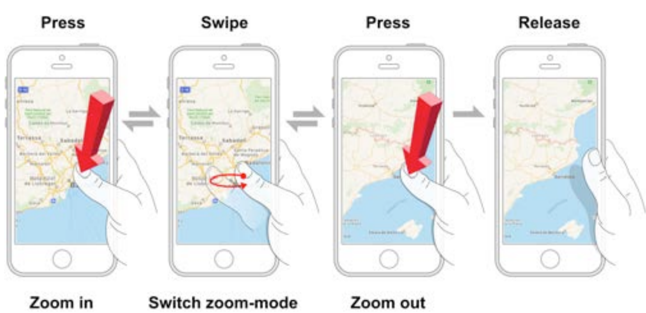

Figure 3: GraspZoom Method

Default ZUI: Double Tap and Slide

This method used the default mobile Google Maps user interface. The user first double-tapped the screen and kept his or her finger on it, and then zoomed by moving the finger up or down. In our implementation, the user moved his or her finger up (down) to zoom out (in).

RESULT: ZOOMING OUT SESSION

Task Completion Time

The average task completion time (in seconds) for the four methods is the result of one trial (one target selection task) (Table 1 and Figure 4). First of all, we conducted a Bartlett's test on average time and result did not show a violation of homogeneity of variances ($X^2(3) = 6.47$, $p = .09$). Then we conducted one-way analysis of variance (within-subjects design; independent variable: type of zooming-out method, dependent variable: task completion time). The result showed a significant difference [$F(3, 92) = 10.51$, $p < .01$, $\eta^2 = 0.26$]. A *post hoc* Tukey's test showed that GraspZoom was significantly slower than Bounce Back ($p < .01$), Force Zoom ($p < .01$) and Default ZUI ($p < .01$).

	Bounce Back		Force Zoom		GraspZoom		Default ZUI	
	mean	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*
Out	8.65	2.70	7.35	3.02	11.35	3.49	7.16	2.04
Out/In	10.50	3.95	9.12	1.46	16.53	4.57	10.90	4.13

Table 1: Average task completion time and standard deviation. Smaller is better.

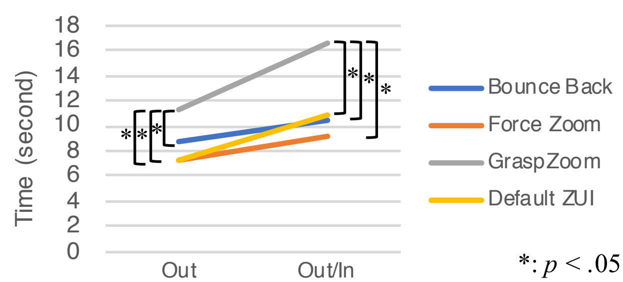

Figure 4: Task completion time.

Number of Operations

The average number of operations for the three zooming-out methods is presented in Table 2 and Figure 5. The numbers of four types of operation were recorded: 1) Bounce Back + double tapping, 2) Force Zoom + double tapping, 3) Default ZUI + double tapping, and 4) GraspZoom. We conducted a Bartlett's test on average number of operations and result did not show a violation of homogeneity of variances ($X^2(3) = 2.45$, $p = .48$). Then we conducted one-way analysis of variance (within-subjects design; independent variable: type of zooming-out method, dependent variable: number of operations). The result showed a significant difference [$F(3, 92) = 17.76$, $p < .01$, $\eta^2 = 0.37$]. A *post hoc* Tukey's test showed that Force Zoom was significantly smaller than Default ZUI ($p < .01$), Bounce Back ($p < .05$), and GraspZoom ($p < .01$). Bounce Back was significantly smaller than GraspZoom ($p < .01$).

	Bounce Back		Force Zoom		GraspZoom		Default ZUI	
	mean	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*
Out	23.79	8.95	16.75	10.56	34.96	8.78	30.42	7.62
Out/In	40.21	8.67	34.13	5.24	58.38	12.91	53.25	22.77

Table 2: Average number of operations and standard deviation. Smaller is better.

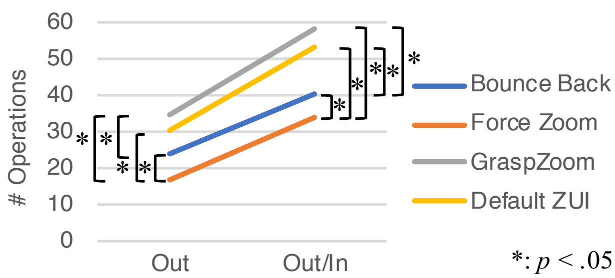

Figure 5: Average number of operations.

	Bounce Back		Force Zoom		GraspZoom		Default ZUI	
	mean	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*	*mean*	*s.d.*
Out	55.10	19.73	69.69	18.26	36.46	19.66	62.50	19.66
Out/In	55.42	18.34	69.48	15.34	28.75	15.61	70.21	17.12

Table 3: System Usability Scale (SUS) score with standard deviation. Larger is better.

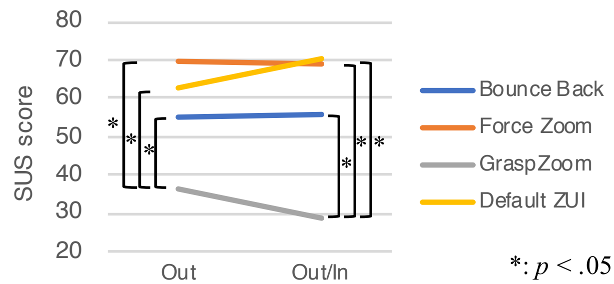

Figure 6: System Usability Scale (SUS) score.

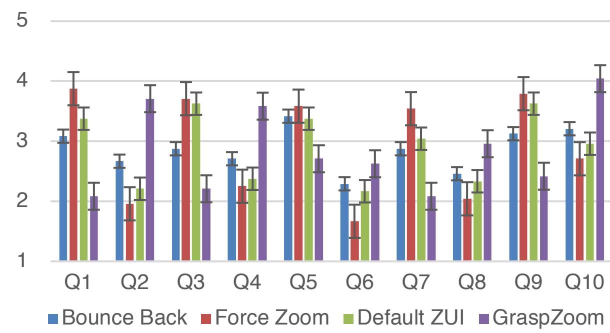

Figure 7: Details of SUS questionnaire result in Zooming out session. Details are presented in the supplementary material.

Usability

We administered a 10-item questionnaire after the completion of each method. A System Usability Scale (SUS) [2], which is a 5-point Likert scale questionnaire, was used. SUS scores were shown in Table 3 and Figure 6. These SUS scores were analyzed using a Friedman test (within-subjects design; independent variable: type of zooming-out method, dependent variable: SUS score). The result showed a significant difference [$X^2(3) = 34.31$, $p < .01$]. A *post hoc* Wilcoxon test with Bonferroni correction showed that Force Zoom was significantly higher than Bounce Back ($p < .01$, $r = 0.488$), and GraspZoom ($p < .01$, $r = 0.627$). GraspZoom was significantly lower than BounceBack ($p < .01$, $r = 0.577$) and Default ZUI ($p < .01$, $r = 0.556$). In general, Force Zoom's SUS score of 69.7 and Default ZUI's SUS score of 62.5 is recognized as Good [1]. Bounce Back received a score of 55.1, which is recognized as OK to below average. GraspZoom's SUS score of 38.0 is recognized Poor.

In terms of individual questions (Figure 7), overall trend of results was that the GraspZoom had significantly lower score than the other three methods. Force Zoom and Default ZUI had similar score in each questionnaire items. Bounce Back was always on the third place. In general, Bounce Back, Force Zoom, and Default ZUI was recognized as a simple (Q2) and easy (Q3), well-integrated interface (Q5) on mobile interaction. They felt confident to user the interfaces (Q9). Details are presented in a supplementary material.

RESULT: ZOOMING OUT/IN SESSION

Task Completion Time

The average task completion time (in seconds) for the four methods is the result of one trial (one target selection task) (Table 1 and Figure 3). First of all, we conducted a Bartlett's test on average time and result showed the violation of homogeneity of variances ($X^2(3) = 26.14$, $p < .01$) so that these average times were analyzed using a Friedman test (within-subjects design; independent variable: type of zooming-out method, dependent variable: task completion time). The result showed a significant difference [$X^2(3) = 30.95$, $p < .01$]. A *post hoc* Wilcoxon test with Bonferroni correction showed that GraspZoom was significantly slower than Bounce Back ($p < .01$, $r = 0.89$), Force Zoom ($p < .01$, $r = 1.0$) and Default ZUI ($p < .01$, $r = 0.72$).

Number of Operations

The average number of operations for the three zooming-out methods is presented in Table 2 and Figure 4. We conducted a Bartlett's test on average number of operations and result showed the violation of homogeneity of variances ($X^2(3) = 48.17$, p < .01) so that these average number of operations were analyzed using a Friedman test (within-subjects design; independent variable: type of zooming-out method, dependent variable: number of operations). The result showed a significant difference [$X^2(3) = 41.53$, $p < .01$]. A *post hoc* Wilcoxon test with Bonferroni correction showed that Force Zoom was significantly smaller than Default ZUI ($p < .01$, $r = 0.83$), Bounce Back ($p < .05$, $r = 0.58$), and GraspZoom ($p < .01$, $r = 0.88$). Bounce Back was significantly smaller than Default ZUI ($p < .05$, $r = 0.62$) and GraspZoom ($p < .05$, $r = 0.79$).

Usability

SUS scores in this session were also shown in Table 3. These SUS scores were analyzed using a Friedman test (within-subjects design; independent variable: type of zooming-out method, dependent variable: SUS score). The result showed a significant difference [$X^2(3) = 37.945$, $p < .01$]. A *post hoc* Wilcoxon test with Bonferroni correction showed that Force Zoom was significantly higher than Bounce Back ($p < .01$, $r = 0.47$), and GraspZoom ($p < .01$, $r = 0.657$). GraspZoom was significantly lower than BounceBack ($p < .01$, $r = 0.611$) and Default ZUI ($p < .01$, $r = 0.635$). Mean SUS questionnaire items are shown in Figure 8. Overall trend of result was similar to Zooming out session.

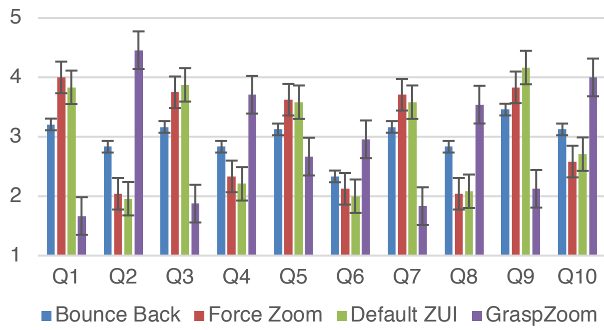

Figure 8: Details of SUS questionnaire result in Zooming out/in session.

DISCUSSION

Overall, all participants were able to successfully use Bounce Back and Force Zoom. The study results confirmed the following:

- Although all participants were using Bounce Back and Force Zoom for the first time, performance was not significantly different from Default ZUI. In contrast, performance was significantly lower on GraspZoom than the other three methods.
- Number of operations was significantly smaller on Bounce Back and ForceZoom than Default ZUI and GraspZoom.
- The usability of GraspZoom was evaluated as significantly lower than Force Zoom, Bounce Back, and Default ZUI. Participants perceived Force Zoom to be easy, and they were confident in using it.

We conducted a simple semistructured interview after the study. Results showed that the users' impressions of GraspZoom were not good. Major reasons given by the participants were *"It was difficult to know the current mode of the interface,"* and *"I need to slide my finger on the screen to know the current mode."* Our simple observation of users' behavior, not small number of participants failed to switch mode on GraspZoom method due to the visual effect is not clear. Likewise, pressure intensity was used to zoom both in and out, so users could understand current mode after they slide their finger on the screen. A finger slide for switching mode of zooming-in and -out was also problematic; we believe that such an interaction state model for mode switch might make users confused. Simple observation of users' behavior in the study, they prefer to use double tapping for zooming-in even in Default ZUI method instead of using sliding action for zooming-in. This implies that users prefer to use different zooming method for zooming-in and zooming-out.

Force Zoom was clearly received positively: *"Force Zoom was good because I do not need to make finger actions repeatedly"* and *"I need to move my finger a lot with Default ZUI, but not with Force Zoom."* As shown by the results for the number of operations, Force Zoom required a very small number of operations to zoom out. Some females commented, *"My hands are small, so I need to repeat finger actions many times,"* and *"My finger got tired."* One interesting comment was as follows: *"I use my smartphone in the bath, so I need to hold it very carefully. Problem is only one hand is available to use it because the other hand is wet while taking the bath. Most importantly, I cannot drop it in the bath, so I do not want to take my finger off the screen."* When using Default ZUI, in addition to repeating actions, users must make larger finger movements, including lifting their finger from the screen. In contrast, the Force Zoom interface requires very small finger movements (pressing down on the screen to zoom out and double tapping to zoom in). Another participant commented, *"I prefer to use Force Zoom while I am lying down on the sofa, because my smartphone usually drops from my hand and hits my face."* Bounce Back got small number of feedbacks. *"Adjustment of zoom level was difficult."*

Limitations

Bounce Back and Force Zoom are proof-of-concept implementations, so there are limitations to the user interface design and results. For example, we did not compare our interfaces with two-handed zooming user interfaces (e.g., pinching). We are interested in comparing the performance of one- and two-handed mobile interaction; however, our interfaces can coexist within one application with a standard zooming user interface, as well as with current practical applications. This work demonstrates one possibility for extending media-browsing interfaces to mobile devices. According to the results of the SUS questionnaire, Force Zoom was well integrated into the standard navigation interface. The simple interviews and behavioral observations showed that using Force Zoom to zoom out integrated well with using the double tap to zoom in.

CONCLUSION

We herein presented the Bounce Back and Force Zoom interfaces, which utilize pressure-sensitive multitouch technology to zoom out. These interfaces were designed to support one-handed use of a mobile device. We conducted a user study to investigate the efficiency and usability of the interfaces by comparing previous research and mobile Google Maps interface as a baseline. Results showed that Force Zoom and mobile Google Maps interface received similar SUS scores from users. Likewise, the task completion time as a performance measure was similar among Bounce Back, Force Zoom, and default mobile Google Maps interface. Number of operations in Bounce Back and Force Zoom was significantly smaller than the default mobile Google Maps interface and previous research, GraspZoom. Results of the SUS questionnaire suggest that the Bounce Back and Force Zoom interface were recognized as a simple and easy interface and can be easily integrated into current mobile devices.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research(A)) Grant Number JP16H02056.

REFERENCES

1. Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: adding an adjective rating scale. *Journal of Usability Studies 4*, 3 (May 2009), 114-123.
2. John Brooke. 1996. SUS: A "quick and dirty" usability scale. *Usability Evaluation in Industry 189*, 194: 4-7.
3. Lei Dong, Carolyn Watters, and Jack Duffy. 2005. Comparing two one-handed access methods on a PDA. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI '05)*. ACM, New York, NY, USA, 235-238. http://dx.doi.org/10.1145/1085777.1085819
4. Edward C. Clarkson, Shwetak Naran Patel, Jeffrey S. Pierce, Gregory D. Abowd. 2006. Exploring Continuous Pressure Input for Mobile Phones. In *GVU Technical Report; GIT-GVU-06-20*. http://hdl.handle.net/1853/13138
5. Seongkook Heo and Geehyuk Lee. 2011. Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 113-122. https://doi.org/10.1145/2037373.2037393
6. Seongkook Heo and Geehyuk Lee. 2011. Force gestures: augmenting touch screen gestures with normal and tangential forces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM, New York, NY, USA, 621-626. https://doi.org/10.1145/2047196.2047278
7. Chris Harrison and Scott Hudson. 2012. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 3149-3152. http://dx.doi.org/10.1145/2207676.2208730
8. David Holman, Andreas Hollatz, Amartya Banerjee, and Roel Vertegaal. 2013. Unifone: designing for auxiliary finger input in one-handed mobile interactions. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 177-184. http://dx.doi.org/10.1145/2460625.2460653
9. Sven Kratz, Ivo Brodien, and Michael Rohs. 2010. Semi-automatic zooming for mobile map navigation. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (MobileHCI '10)*. ACM, New York, NY, USA, 63-72. https://doi.org/10.1145/1851600.1851615
10. Sylvain Malacria, Eric Lecolinet, and Yves Guiard. 2010. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2615-2624. https://doi.org/10.1145/1753326.1753724
11. Takashi Miyaki and Jun Rekimoto. 2009. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, , Article 11 , 4 pages. http://dx.doi.org/10.1145/1613858.1613872
12. Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. 2006. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services (MobileHCI '06)*. ACM, New York, NY, USA, 203-210. http://dx.doi.org/10.1145/1152215.1152260
13. Keith B. Perry and Juan Pablo Hourcade. 2008. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics Interface 2008 (GI '08)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 57-64.
14. Ken Hinckley and Hyunyoung Song. 2011. Sensor synaesthesia: touch in motion, and motion in touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 801-810. https://doi.org/10.1145/1978942.1979059
15. Jun Rekimoto. 1996. Tilting operations for small screen interfaces. In Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96). ACM, New York, NY, USA, 167-168. http://dx.doi.org/10.1145/237091.237115

16. Jun Rekimoto and Carsten Schwesig. 2006. PreSenseII: bi-directional touch and pressure sensing interactions with tactile feedback. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1253-1258. http://dx.doi.org/10.1145/1125451.1125685
17. Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*, 146–153. http://doi.acm.org/10.1145/1385569.1385594
18. Craig Stewart, Michael Rohs, Sven Kratz, and Georg Essl. 2010. Characteristics of pressure-based input for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 801-810. https://doi.org/10.1145/1753326.1753444
19. Ryosuke Takada, Wei Lin, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. 2017. A Technique for Touch Force Sensing using a Waterproof Device's Built-in Barometer. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2140-2146. https://doi.org/10.1145/3027063.3053130