TouchShield: A Virtual Control for Stable Grip of a Smartphone Using the Thumb

Jonggi Hong
HCI lab
KAIST
291 Daehak-ro, Yuseong-gu,
Daejeon 305-701, Korea
koreahjg@kaist.ac.kr

Geehyuk Lee
HCI lab
KAIST
291 Daehak-ro, Yuseong-gu,
Daejeon 305-701, Korea
geehyuk@gmail.com

Copyright is held by the author/owner(s).
CHI 2013 Extended Abstracts, April 27–May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract
People commonly manipulate their smartphones using the thumb, but this is often done with an unstable grip in which the phone lays on their fingers, while the thumb hovers over the touch screen. In order to offer a secure and stable grip, we designed a virtual control called TouchShield, which provides place in which the thumb can pin the phone down in order to provide a stable grip. In a user study, we confirmed that this form of control does not interfere with existing touch screen operations, and the possibility that TouchShield can make more stable grip. An incidental function of TouchShield is that it provides shortcuts to frequently used commands via the thumb, a function that was also shown to be effective in the user study.

Author Keywords
TouchShield; safety; function keys; touch screen smartphone

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): User Interfaces – Input devices and strategies.

Introduction
Nowadays, people use smartphones for many kinds of tasks in various situations. For example, they check e-mail while walking and watch video clips while standing

a) Reading contents

b) Manipulating controls

c) Holding a small-screen device

Figure 1: Grips for different states and for different devices.

in a bus. In many other common situations such as these, they operate their smartphones with one hand, while the other hand is engaged in another task, such as carrying a bag or holding a handle [3]. Operating a smartphone with one hand can be unsafe, especially when the smartphone is large. Figure 1a shows a typical hand position when a user is reading screen contents, and Figure 1b shows a typical position when the user is interacting with the screen contents. Though users may switch between these two positions, as the contents become more interactive, they tend to stay in the second position. This second position, however, is unstable, as the phone lies on the fingers while the thumb hovers over the touch screen. Manipulating a smartphone in such an unstable position poses a usability problem, especially in a mobile situation. In response to this difficulty, smartphone accessories for stable one-handed operation are being introduced in the market [8].

This usability problem is, in fact, a new phenomenon that arose with the introduction of smartphones with large touch screens. A typical hand grip before these large-screen devices were introduced is shown in Figure 1c. While users interacted with these phones with their thumb, there was a place for the thumb to rest on. This place usually took the form of a circle or a square, and consisted of keys for frequently used commands such as [Ok], [Back], or [Menu]. This design feature, with a function key set under the thumb, persisted for a long time, beginning in the days of the clamshell phone, and on into the days of the slider phone. This kind of grip shown in Figure 1c even persisted into the early days of the smartphone. Then, with the introduction of large-screen smartphones such as the Apple iPhone, this design feature disappeared. The function key set provided a place for the thumb, making a stable grip possible. In addition, it offered efficient shortcuts to frequently used commands.

In order to make a stable grip possible while manipulating a smartphone with the thumb, we designed a virtual control called TouchShield, which provides a resting place for the thumb in much the same way as the earlier physical function key set did. In a user study, we observed that TouchShield provided participants with a stable grip, and that it did not interfere with existing touch screen operations. Since TouchShield's design was based on the physical function key set, it also provides shortcuts to frequently used commands underneath the thumb, as its physical counterpart did. Although this is an incidental function of TouchShield that is not the focus of this paper, the benefit of this function was also demonstrated in our user study.

Related Work

It is not easy to hold a smartphone stably, because the large touch screen makes the front side of the phone inaccessible to one's grip. In response to this problem, a number of companies are introducing smartphone accessories that help stabilize one's grip [8]. However, these products require the addition of physical parts external to the device, which many find unacceptable for aesthetic reasons. The fact that no satisfactory solution to this problem has yet been found was our primary motivation in developing TouchShield.

When users hold and control a smartphone using one hand, the key muscles of the hand must be used to grasp the device. For this ergonomic reason, users have no choice but to use only the thumb to control the

a) Intermediate state

b) TouchShield state

c) Executing a function (back)

Figure 2: Activation of TouchShield and its function keys.

touch screen [3]. Moreover, the movement of the thumb is limited, and as the size of these devices grows, it is becoming increasingly difficult to reach the entire area of the touch screen [5]. Moreover, the proficiency needed to select a target on a touch screen is different for different locations on the screen and for different sizes of target [7]. It is difficult to use hardware buttons and keys below a touch screen without changing one's grip. As a result, users have to frequently change their grip, and they experience unstable control of the mobile device [6].

Because the contact size is relatively easier to obtain from a touch screen than other properties, it has been used to enrich the input techniques for a touch screen. For example, *SimPress* provides a hovering operation that is based on the concept that a touch with a small contact size is a hovering operation [1]. In addition, *Fat thumb* implemented the concept that the contact size can be used to select different control modes, and applied this concept to the pan and zoom control [2].

Design

In our design, when a user is controlling a smartphone as usual, TouchShield is not activated. However, when TouchShield detects a large contact area, a circle progressively fades in to indicate the activation progress (Figure 2a). If the user continues to touch , TouchShield is fully activated and virtual keys appear around the thumb (Figure 2b). Even when it is fully activated, TouchShield is not completely opaque, so users are able to read the contents underneath TouchShield. TouchShield has icons on four sides. When the user slides his/her thumb toward an icon, an assigned function is triggered (Figure 2c). After such a function is triggered, or when the user releases his/her fingers, TouchShield returns to its transparent state. In the latter case, TouchShield does not disappear instantly; it fades out. While TouchShield is fading out, the user can re-activate it by simply touching it again.

Implementation

At first, the implementation of TouchShield seemed to be straightforward. Contrary to our expectation, though, we had to address many technical issues in order to arrive at the current prototype. Next, we explain the two major issues that we had to deal with.

Avoiding interference with existing actions

Users sometimes use a large touch area in ordinary operations. This does not happen often in selection operations, but it happens frequently in gesture operations such as scrolling and pinching. For this reason, it was necessary to add an intermediate state before the activation of TouchShield, so that its activation can be canceled in the case of a gesture operation. Since it is also desirable for users to have a chance to recover from an accidental release, we added another intermediate state before TouchShield is completely deactivated.

Figure 3 shows these state transitions of TouchShield. In the Normal state, users can use the touch screen as usual. When TouchShield detects a large contact size, it moves to the Intermediate state. If the user maintains contact for 350 ms in the Intermediate state, TouchShield moves to the TouchShield state, in which it is fully activated. In this state, users can use the function keys by dragging. TouchShield moves to the Restoring state if the user releases his/her finger in the TouchShield state. This state is also an intermediate state and is only maintained briefly. After this state has

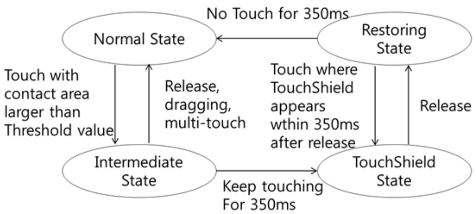

Figure 3: State transition diagram of TouchShield.

expired, TouchShield returns to the Normal state and becomes transparent. If the user touches TouchShield during the Restoring state, TouchShield returns to the TouchShield state immediately.

In order to avoid unintended TouchShield activation, we allowed it to be cancelled when the thumb moves more than a certain distance in the Intermediate state. We conducted a pilot study to determine both the relevant distance and time threshold. In the pilot study, participants were asked to read a news article and activate TouchShield. In more than 90% of the cases of TouchShield activation, the touch distance was shorter than 30 px, which is 2.7 mm on a Samsung Galaxy Note GT-N7000. We determined the duration of the Intermediate state by measuring the time it took to scroll more than 30 px on the news article. It took less than 350 ms when participants scrolled quickly, and 300 ms to 1000 ms when they scrolled while reading the content. Taking all these considerations into account, we chose 350 ms as the duration time of the Intermediate state. This allows sufficient time to detect a quick scroll, while a slow scroll will not be detected. As we observed in our pilot study and evaluation study, this seldom caused a problem, because a large contact area rarely occurs during slow scrolling.

Accommodating individual differences in contact size
Since people have different-size thumbs, using a fixed threshold value to classify large versus small contact areas is not a reliable approach. We therefore used an adaptation algorithm that can change the threshold based on the user's contact size history. The adaptation algorithm collects the following two types of touch samples.

- **Selection touch**: The touch used for selecting an object while TouchShield is in the Normal state.
- **Holding touch**: The touch used to activate TouchShield, and subsequently, to use a virtual function key.

We chose an initial threshold value of 0.35 based on our experience and observation during the development process. The unit of the touch size is a relative one used by the Android 4.0.3 API. When 10 selection touch samples and 10 holding samples are collected, the threshold value is updated using the formula, $(A_s + A_h)/2$, where A_s and A_h are the average contact sizes of the selection touch and holding touch, respectively. Figure 4 shows an example of variations in the threshold as more touch samples are collected. As can be seen in the figure, the initial threshold was too low for the user, and the modified threshold fits better for classifying the two types of touch.

Figure 4: Example of threshold variation. The final threshold value is 0.54.

Experimental evaluation

The prototype of TouchShield was built on a Samsung Galaxy Note GT-N7000 with a display resolution of 800

× 1280 and a diagonal size of 5.3 inches, running Android 4.0.3. This prototype simulates a web browsing environment on a mobile device. In this prototype, the following functions are assigned to the four virtual function keys: (up) Home, (down) Sleep, (left) Back, and (right) Menu. The device has physical function keys located at the bottom left (Menu), bottom center (Home), and bottom right (Back) sides of the front surface, and on the right side of the device (Sleep). During the test, participants were requested to use only one hand.

We conducted two experiments: a stable grip test and a virtual function key test. The aim of the stable grip test was to estimate the stability of the grip when users used the device both with and without TouchShield. The aim of the virtual function key test was to estimate the performance of the virtual function keys.

Stable grip test

In this experiment, we asked participants to perform two types of tasks. First, participants were asked to open a door and push a button in an elevator while using the device with one hand. We only gave instructions about how to use TouchShield, but not asked to use TouchShield. The goal was to observe how users use TouchShield to obtain a stable grip when they performed multiple tasks at the same time. The participants opened a door thrice and pushed a button to use an elevator thrice. In addition, participants read news articles using our prototype while walking and while running. Five participants were recruited from our university (24 to 30 years old, all male). All participants were regular users of a touch screen phone and all were right-handed.

Virtual function key test

The participants were given a list of six tasks: four tasks for using each of the respective function keys, and two tasks for testing for possible interference with scrolling. (There were actually seven tasks, but the last task was a two-handed task that is not analyzed in this paper.) When participants selected a task between Tasks 1 and 4, a page with a start button appeared, instructing them which function key they should use there. Task 5 was to read a long article that required scrolling in order to reach the end. Task 6 was to follow arrows by scrolling the screen. The arrows pointed down, right, and left, so that participants had to scroll both vertically and horizontally. Using TouchShield, participants performed all six tasks. Using the physical function keys, they only performed the first four tasks. For each technique, participants repeated this process 10 times. Ten participants (17 to 25 years old, all male) were recruited from our university. All participants were right-handed and were regular users of a touch screen phone.

Results

Users' feedback from stable grip test

All participants responded that TouchShield was useful when they were running. However, participants did not need to use TouchShield when they walked because walking and controlling the smartphone with one hand did not produce a particularly unstable grip. One participant said that the presence of TouchShield around the thumb interfered with the screen contents, even though it is translucent. Moreover, the functions mapped to the virtual function keys were often executed accidentally, because the thumb tends to move unintentionally when one is running.

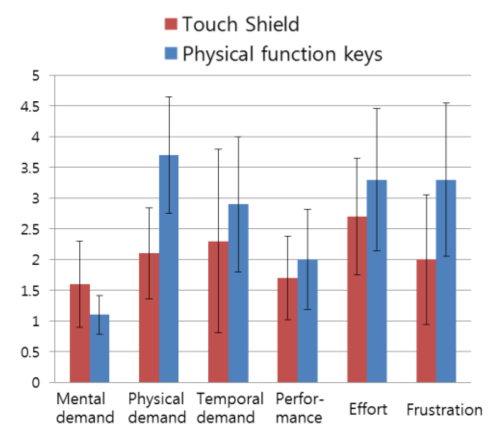

Figure 5: Mean values of NASA TLX scores for each technique.

We also observed whether participants used TouchShield while opening doors and pushing buttons. Under these conditions, three participants actually used TouchShield. Two participants who did not use TouchShield used a scrolling gesture in order to rest their thumb on the touch screen. Thus, although they did not use TouchShield, they inadvertently demonstrated the need for TouchShield.

Interference with existing operations

During the virtual function key test, none of the participants experienced any interference while reading the articles in task 5, or while following the arrows in task 6. Though participants often scrolled the pages using a contact size larger than the threshold value, TouchShield did not interfere with the operations.

Task load

After the virtual function key test, NASA TLX was used to measure the comparative task loads involved in using TouchShield and the physical function keys; the results are shown in Figure 5. Wilcoxon signed-rank test results showed that TouchShield is significantly less physically demanding ($Z = 22.5$, $p < 0.004$). In other aspects (temporal demand, mental demand, performance, frustration, and effort), there was no significant difference between the two conditions ($p > 0.05$).

Conclusion and future work

We designed a virtual control called TouchShield, and through experiments, we could verify its potential as a software solution to make the user's grip of a smartphone more stable, and to make function keys that are easier to access than physical function keys. TouchShield did not interfere with existing touch screen operations, such as scrolling, and could also adjust to individual differences in thumb size via an adaptive algorithm. We are currently working on the next prototype of TouchShield, which will accommodate the user feedback we have received, and resolve the usability issues that arose. Our user study will focus on the situations in which TouchShield is necessary in order to obtain a stable grip. Also, our future research will further investigate a method of distinguishing the user's gripping action from control gestures.

References

[1] Benko, H., Wilson, A.D., Baudisch, P. Precise Selection Techniques for Multi-Touch Screens. Proc. of CHI '06, ACM (2006), 1263-1272.

[2] Boring, S., Ledo, D., Chen, X., Marquardt, N., Tang A., Greenberg S. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. Proc. MobileHCI'12, ACM(2012).

[3] Hirotaka, N. Reassessing Current Cell Phone Designs : Using Thumb Input Effectively. Proc. of CHI EA '03, ACM (2003), 938-939.

[4] Karlson, A.K., Bederson, B.B., Contreras-Vidal, J.L. Understanding single-handed mobile device interaction. Technical Report HCIL-2006-02, Computer Science Dept, University of Maryland at College Park (2006).

[5] Karlson, A.K., Bederson, B.B., Contreras-Vidal, J.L. Studies in One-Handed Mobile Design: Habit, Desire and Agility. Proc. UI4ALL '06.

[6] Pascoe, J., Ryan, N., Morse, D. Using while moving: HCI issues in fieldwork environments. ACM Trans. Comput.-Hum. Interact. 7, 3 (2000), 417-420.

[7] Perry, K.B. and Hourcade, J.P. Evaluating one handed thumb tapping on mobile touchscreen devices. Proceedings of GI'08.

[8] Research in FlyGirip™, FlyGirip™ Technology, http://shop.flygrip.com/