ForceClicks: Enabling Efficient Button Interaction with Single Finger Touch

Sangeon Yong
GSCT, KAIST
Daejeon, South Korea
koragon2@kaist.ac.kr

Liwei Chan
Dept. of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
liweichan@cs.nctu.edu.tw

Edward Jangwon Lee
GSCT, KAIST
Daejeon, South Korea
noshel@kaist.ac.kr

Juhan Nam
GSCT, KAIST
Daejeon, South Korea
juhannam@kaist.ac.kr

Roshan Peiris
Keio Media Design
Yokohama, Japan
roshan@kmd.keio.ac.jp

TEI '17, March 20-23, 2017, Yokohama, Japan
ACM 978-1-4503-4676-4/17/03.
http://dx.doi.org/10.1145/3024969.3025081

Abstract
ForceClicks is a novel touch button input technique for consecutive clicking which incorporates touch force sensors. From force data of a single continuous touch over time, ForceClicks detects peaks and generates discrete clicks. Compared to typical button interaction, this is effective in a sense that consecutive clicks do not require finger positional movements. Additionally, stable force over a certain time threshold can trigger an alternate state, *long press*, and can be mapped to other actions. The usability of ForceClicks has been evaluated in terms of a) scattering level and b) efficiency. Results suggest higher stability than typical touch, especially when the task requires visual engagement on remote content. The relatively scatter-free characteristic of ForceClicks allows it to be applied on rapid clicking while gaming, and reduce of visual dedication allows easier control of external devices, and two applications, a shooting game and a number picker, are presented for demonstration.

Author Keywords
force touch; 3D touch; touch screen; gestures

ACM Classification Keywords
H.5.2 [User Interfaces]: Input devices and strategies (e.g., mouse, touchscreen)

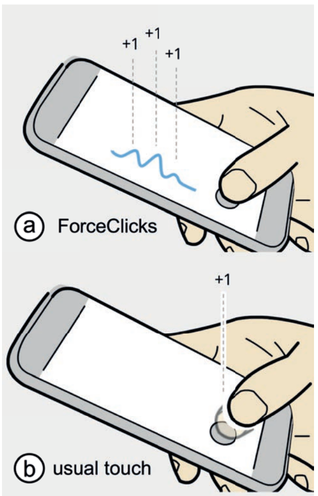

Figure 1: ForceClicks

You can access the video of ForceClicks through the link below.
https://youtu.be/im5fEsX6yHE

Introduction

Touch-activating button is generally considered efficient, but repeating the actions on the same button may cause misses. These misses occur due to increase in touch point scattering over multiple landings. Unlike physical buttons, which provide tangible interfaces and the user is able to haptically ensure every interaction, it is difficult to reduce this scattering as touch-activated buttons do not provide tactile feedback. Missing a button while consecutive tapping often happens when the task requires users' visual dedication at remote content, such as pressing onscreen buttons while concentrating on the entire game screen.

To reduce such scattering and missing problems, in this research we propose *ForceClicks*, a novel button input technique that incorporates force of touch.

ForceClicks

By analyzing force change of a single touch over time, ForceClicks detects peaks of the force curve and recognizes those as discrete clicks. Figure 1 illustrates the difference between ForceClicks and ordinary click. While ordinary clicks require the user to initiate a new touch for every click, ForceClicks can receive consecutive inputs without requiring further finger movement. Additionally, ForceClicks also features *long press*, a mode that is triggered by maintaining constant touch force over a certain amount of time.

Zero physical movement of fingertip brings several benefits compared to ordinary touch buttons. First, scattering of the position of each consecutive touch is reduced. This suggests that users are able to consecutively click on a button without missing even when they are not able to pay visual attention on the button. Second, it reduces the fatigue of finger as less physical movement is required than ordinary clicking.

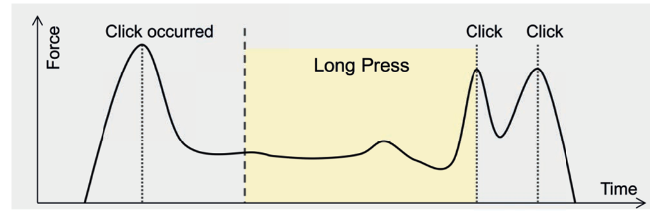

Figure 2: basic mechanism of ForceClicks

Approach

Typical touch force input, such as iPhone's peek-and-pop, sets an absolute force threshold for feature activation and requires the user to practice and adapt their finger force levels to the system. Regarding to setting absolute force thresholds as activation triggers, previous research indicates that proper force of touch control can be done when users are given visual feedback of currently applied force and extended input time. With both conditions provided, users were able to properly home into several intervals without any problems [1, 3]. However, when the input time was reduced to an instant, users showed only 58.5% success rate with six force intervals, compared to 99.9% with two [2]. This suggests that without visual feedback and sufficient input time, users are not able to precisely control the absolute magnitude of touch force. Therefore, in ForceClicks, rather than setting absolute force thresholds we focus on relative force change over time and extract data, rather than absolute force intensity.

The force profile that occurs during a force click is shown in Figure 2. Here, the ForceClicks algorithm observes the rising and falling slopes on the profile and identifies the peak at which a "click" is triggered.

To find peaks in the force values, we define three states

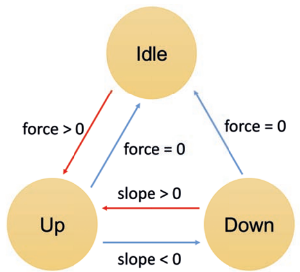

Figure 3: state diagram of ForceClicks

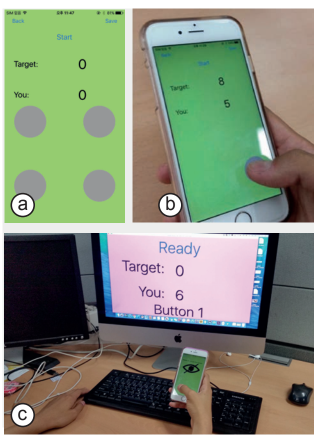

Figure 4: experimental design

of touch: *idle, up,* and *down*. Each state corresponds to the current force change direction. *Idle* state occurs when the button is untouched, that is, zero force. *Up* and *down* states occur when the first time derivative of force changes sign: the system enters up state when the force begins to increase and down state upon decrease. Releasing the button causes the force value to hit zero and the state returns to idle. Counting discrete clicks can be done at either *up to down* or *down to up* transitions, and conducting simple user tests on both settings suggest that counting down to up (or idle to up) transitions produce more accurate clicks.

Although this simple three-state approach with down to up (positive edge) triggering can be used for consecutive discrete clicking without difficulties, incorporating *long press* feature – maintaining a level of force over a certain time – into the system for additional functions requires a more complex input model. For most users, maintaining equal force over time is impossible and unintended peaks of force tend to appear when they try to apply constant force. Therefore, in order to properly parse discrete clicks alongside with long presses from touch force data, distinguishing between intended and unintended peaks is essential.

Long press state is entered when force is held in an even level over a certain amount of time – no intended clicks, or peaks, occur over a time threshold. In order to rule out unintended peaks, two criteria are set based on user observations and force peaks are considered as intended only when both are sufficed. First, intended clicks must have a short up state. User observations suggest that most clicks are done in a swift manner, therefore up slopes longer than 30 frames (0.5 seconds) are set to be regarded as unintended clicks. Second, the second time derivatives of force during an intended click must have a wide range: difference between the maximum and minimum value must be larger than a certain threshold. The lifespan of a click is defined as follows: a click starts at the beginning of an up state and ends when down state begins and a) continues more than 5 frames, b) another up state starts before 5 frames, and c) user removes touch and enters idle state within 5 frames. A wider range of second derivatives in an up slope implies that an intended click should start and end with higher acceleration, and 0.2 has been set as the minimum range threshold of an intended click.

User Study

This study explores the performance of ForceClicks in terms of scattering level and efficiency regarding existing touch-activating buttons (e.g., TouchClicks).

Experimental Design

The study was a within-subject design with two variables, operation and vision. The operation variable includes ForceClicks or TouchClicks conditions. The vision variable describes whether the participants' vision is engaged with the input control or with the remote content, i.e., vision-on-control or vision-off-control conditions. All tests were implemented on Apple's iPhone 6s. For each trial, the participant was instructed to reach a randomly assigned number of clicks (between 8-12) on 1 of 4 buttons displayed on the phone screen with the assigned interface conditions, as shown in Figure 4a. The number of clicks being performed was displayed on the phone screen in vision-on-control condition (Figure 4b), and on the remote screen in vision-off-control condition (Figure 4c). In vision-off-control condition, the phone screen turned black after the participants performed the first click. Button positions were randomized, and both interface conditions were counter-balanced. The Clicks were repeated for 10 times for each button position. In total, each participant performed 2x2x4x10 = 160 trials.

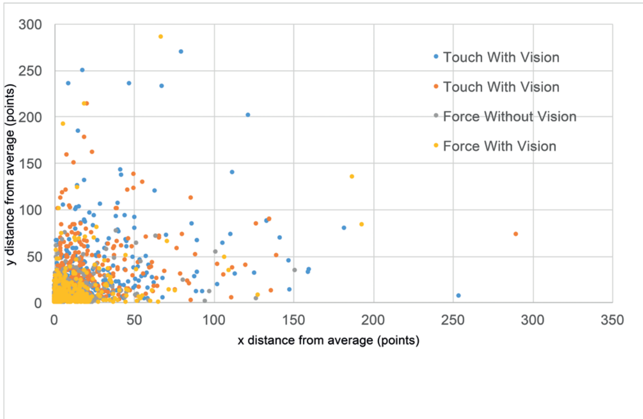

Figure 5: Variance distribution of four test conditions.

Ten participants (8 males and 2 females, all right-handed, with average age 27) were recruited. All participants used smart phones. After being introducted to the study, they received a 5-min training session, going through all conditions.

Result Discussion
Figure 5 shows the position accuracy results of four conditions. The scattered colors manifest that the variance of position decreases When using ForceClicks. On average, the position variance is doubled when participants use the touch-activating buttons (23.117 to 55.228), and the vision does not affect the position accuracy (40.315 to 38.030). On the other hand, the time consumption of both conditions are similar (0.189 to 0.173). This result can show the scatter-free property of ForceClicks. Also, most participants mentioned that ForceClicks is easy to learn, and less fatiguing to the finger.

Figure 6: Space shooting game implemented with ForceClicks.

Example Applications
Two example applications have been implemented to demonstrate how ForceClicks can be used in real-world applications. The first application is a simple shooting game, and the second is a number picker. We implemented them by utilizing both discrete clicks and long press in ForceClicks.

Space Shooting Game
Figure 6 shows the simple space shooting game to demonstrate the advantage of ForceClicks. On the left part of screen, users can control the spaceship's location. On the right, a ForceClicks button is placed instead of ordinary touch buttons for gunfire. Discrete clicks fire individual missiles, and long press mode fires a continuous laser weapon. Due to the less-scattering characteristic of ForceClicks, users can fire weapons more easily while visually paying attention to targets such as enemy vessels and obstacles.

Number Picker
To test the usability of the long press mode, we implemented a number picker using ForceClicks. Discrete clicks on the button increase the number by one. Entering the long press mode increases the number continuously, and the increasing speed can be dynamically changed by controlling touch force: greater force speeds up the increase. This number

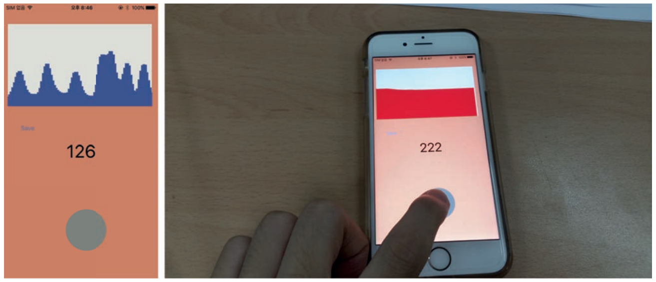

Figure 7: Number Picker implemented with ForceClicks.

picking can be efficient in a sense that the finger can stay on the button until it reaches the target number while controlling the speed of number change.

Conclusion

ForceClicks is a novel button interface for pressure-sensitive touch screens that do not require releasing the finger, fixed force threshold, or haptic feedback for consecutive clicks. While ForceClicks buttons look and feel identical to ordinary buttons, enabling the touching fingertip to rest on the button during consecutive cliks offer two benefits: a) less scattering and b) less fatigue. We performed an experiment to compare ForceClicks and ordinary touch buttons and received positive feedbacks in efficiency and usability. Additionally, two sample applications were implemented:a) space shooting game and b) number picker, to demonstrate the practical usage of ForceClicks.

Future Work

ForceClicks is an on-going project and we are actively seeking and working on further potential improvements. First, although a single ForceClick button may be placed anywhere on the screen, under multitouch situations iPhone force sensors have a problem that the force readings of nearby touches interfere with each other. This acts as a design restriction: ForceClick buttons should be placed far enough. We are seeking for a software workaround to overcome this. Second, our force thresholds for filtering out unintended clicks rely on user observations and parameter tunings. We plan to fine-tune the parameters by employing more statistical data into our model. Finally, long press also requires improvement. Currently, long press mode triggers upon a constant level of force over a fixed time threshold, and this disables interaction with the button for a certain amount of time. For example, in the space shooter game, continuous lasers can be fired only after a pause in shooting discrete missiles. Ideas that can reduce this threshold will surely improve ForceClicks.

REFERENCES

1. Sachi Mizobuchi, Shinya Terasaki, Turo Keski-Jaskari, Jari Nousiainen, Matti Ryynanen, and Miika Silfverberg. 2005. Making an impression: force-controlled pen input for handheld devices. In *CHI'05 extended abstracts on Human factors in computing systems*. ACM, 1661–1664.

2. Esben Warming Pedersen and Kasper Hornbæk. 2014. Expressive touch: studying tapping force on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 421–430.

3. 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. DOI:http://dx.doi.org/10.1145/1851600.1851631