LETTER

Design and Feasibility Study: Customized Virtual Buttons for Electronic Mobile Devices

Seungtaek SONG†, **Namhyun KIM**†, ***Nonmembers***, **Sungkil LEE**†, **Joyce Jiyoung WHANG**†, ***Members***, ***and*** **Jinkyu LEE**†a), ***Nonmember***

SUMMARY Smartphone users often want to customize the positions and functions of physical buttons to accommodate their own usage patterns; however, this is unfeasible for electronic mobile devices based on COTS (Commercial Off-The-Shelf) due to high production costs and hardware design constraints. In this letter, we present the design and implementation of customized virtual buttons that are localized using only common built-in sensors of electronic mobile devices. We develop sophisticated strategies firstly to detect when a user taps one of the virtual buttons, and secondly to locate the position of the tapped virtual button. The virtual-button scheme is implemented and demonstrated in a COTS-based smartphone. The feasibility study shows that, with up to nine virtual buttons on five different sides of the smartphone, the proposed virtual buttons can operate with greater than 90% accuracy.

key words: *virtual buttons, mobile devices, accelerometer, gyroscope, design and feasibility study*

1. Introduction

Individual users exhibit different usage patterns with electronic mobile devices such as smartphones and tablet computers, which require tailored physical buttons. For example, a user who frequently uses a social media service may wish to have a shortcut button to the service, while another user may require a physical button for screenshots. However, such individual demands cannot be addressed in COTS-based (Commercial Off-The-Shelf-based) devices due to high production costs and hardware design constraints.

The goal of this paper is to introduce a framework for using multiple virtual buttons, in which positions and functions are customized by individual users. Targeting electronic mobile devices equipped with an accelerometer and gyroscope (such as smartphones), we present the design and implementation of the framework, which does not require deployment of additional hardware. To this end, we develop sophisticated strategies firstly to detect when a user taps one of the virtual buttons, and secondly to locate the position of the tapped virtual button. We implement the first strategy using the accelerometer; a tap event is detected by monitoring whether the sensor's readings exceed a predefined threshold. For the second strategy, we utilize the accelerometer to detect on which side a user taps, and the gyroscope to detect the exact position at which a user taps on the given side.

Manuscript received July 19, 2018.
Manuscript revised November 29, 2018.
†The authors are with College of Software, Sungkyunkwan University (SKKU), Republic of Korea.
a) E-mail: jinkyu.lee@skku.edu (Corresponding author)
DOI: 10.1587/transfun.E102.A.668

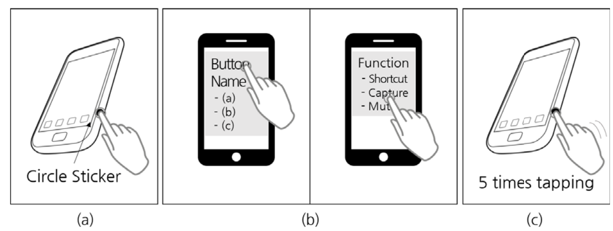

Fig. 1 Setup of customized virtual buttons.

The configuration of the proposed virtual-button framework is shown in Fig. 1. First, a user marks the position of a virtual button they wish to deploy (Fig. 1(a)). Second, the user maps the position to the name and preferred action when the button is tapped, e.g., launching the social media application (Fig. 1(b)). Finally, the user initializes the button by tapping it five times (Fig. 1(c)). Once a user finishes the three configuration steps, s/he can use the virtual button by tapping the corresponding position once per request, as if there were a physical button at the corresponding position.

2. Recognition of Tap Event

To develop a framework of virtual buttons, the first step is to recognize whether a user taps one of the virtual buttons. To this end, we investigate the change in acceleration when a user taps on a right side of a Nexus 5 smartphone, using accelerometer readings. As shown in Fig. 2(a), a right-side tap yields an abrupt change in the accelerometer x-axis readings, while the y and z axes do not change significantly. Therefore, once we have carefully determined a threshold for the acceleration value, we can recognize a tap event. As shown in Fig. 2(a) for instance, a threshold for $-7\,\mathrm{m/s^2}$ (the red line) can be an appropriate threshold for detection of a tap event.

There might be a misinterpretation when a user moves (e.g., walking up or down stairs), because such movement yet without tapping events may also yield non-trivial changes in acceleration. Hence, it is important to distinguish true tap events and no-tap events (but with user movements) to avoid false positives. For this, we compared a stationary tap event against a no-tap event with movement, and identified their significant differences in terms of acceleration rate changes. The former yields a significant change with even one interval (for example, a $7\,\mathrm{m/s^2}$ change during 10 ms as

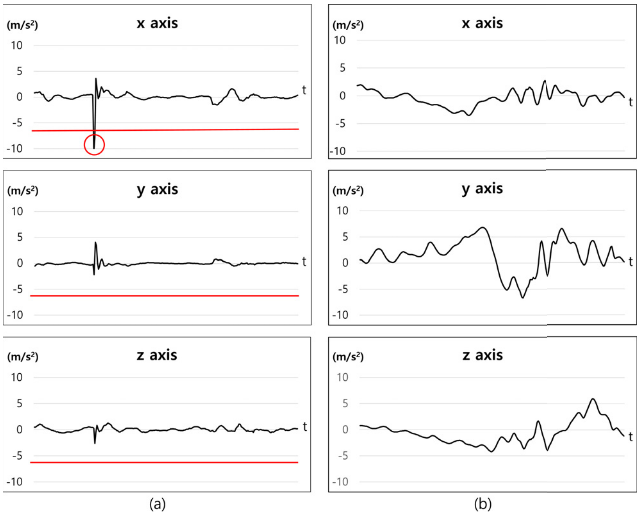

Fig. 2 Accelerometer readings every 10 ms: (a) a tap event without any movement; and (b) no tap event when moving down the stairs.

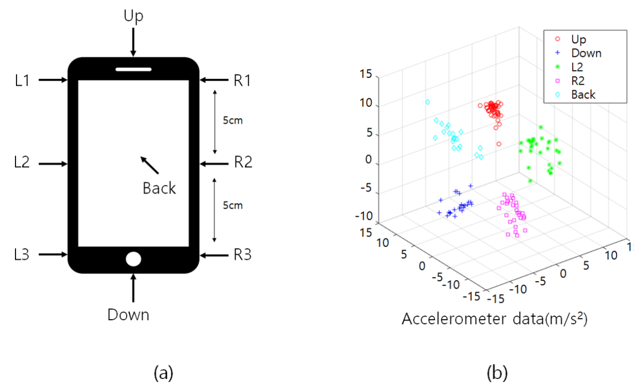

Fig. 3 (a) Positions of nine virtual buttons; and (b) accelerometer data readings (20 taps for each side).

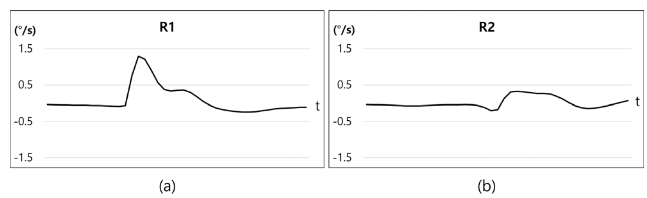

Fig. 4 Change in the gyroscope x-axis readings when a user taps the positions of (a) R1 and (b) R2.

shown in Fig. 2(a)), while the latter does not (for example, no more than a 2 m/s^2 change during 10 ms). In other words, a user who moves around does not yield an accelerometer change as suddenly as a tap event without any movement. This implies that we also need to determine a threshold for the rate of the accelerometer change; for example, in Fig. 2 we can recognize the tap event if the amount of acceleration change for one of the three axes, is more than 7 m/s^2 during 10 ms.

3. Position Detection of Tap Event

When a tap event has been recognized, the next step is to detect the position at which the tap event occurred. To this end, we target the nine virtual smartphone buttons shown in Fig. 3(a), and investigate acceleration and angular acceleration changes in the accelerometer and gyroscope readings, respectively. We first focus on the readings of the three-axis accelerometer. As shown in Fig. 2(a), each tap event leads to a sudden change in only one of the three axes. If a user taps the left/right, top/bottom, or back of the smartphone, there is a significant acceleration change in the x, y, and z axis, respectively; for example, Fig. 2(a) represents a right-side tap. The sign of the acceleration change depends also on the position. Left and top taps result in positive values for the x and y axis, respectively, while those for the right side and bottom result in a negative value for the corresponding axis. The side tapped is therefore distinguished by the sign and position of acceleration change, as shown in Fig. 3(b). We can apply the same method to recognition of the tap event, which is based on the threshold for the rate of acceleration change. That is, we first find an axis for which the acceleration change is larger than the threshold for a given time unit (for example, 7 m/s^2 during 10 ms). We can then recognize the tap as a left, right, top, bottom, or back tap, if the combination of axis and acceleration change is (x,+), (x,-), (y,+), (y,-), or (z,-), respectively.

The next step is to determine the exact location of a tap on a given side. To find this, we use the angular acceleration data from gyroscope readings. Let us compare the three situations where a user taps the R1, R2, and R3 positions shown in Fig. 3(a). A tap to R1 results in a counter-clockwise rotation of the smartphone, yielding the positive x-axis angular acceleration change shown in Fig. 4(a). A tap to R2 creates little rotation, yielding the small angular acceleration change shown in Fig. 4(b). The x-axis angular acceleration change when R3 is tapped shows a negative change, in contrast to the change when R1 is tapped. Therefore, if we focus on the angular acceleration change, we can locate the position of a tap on the right side. Likewise, it is possible to distinguish the tapped position within the left side, for example L1, L2, or L3 in Fig. 3(a), in a similar way.

In summary, we can detect on which side a tap occurs by using accelerometer readings, and can then distinguish the exact position within the side using gyroscope readings.

4. Implementation and Evaluation

We implemented the proposed framework using a Nexus 5 smartphone with nine virtual buttons (Fig. 3(a)), and used the LIBSVM library [1] (with linear kernel) to classify the tapped position. We then evaluated the accuracy of the proposed framework. Ten paid subjects participated in the experiment, in the age range 20 to 70, and with four females and six males.

As we described in Sect. 3, we can construct a feature vector for each tap event by reading the data from accelerometer and gyroscope. Since there are predefined la-

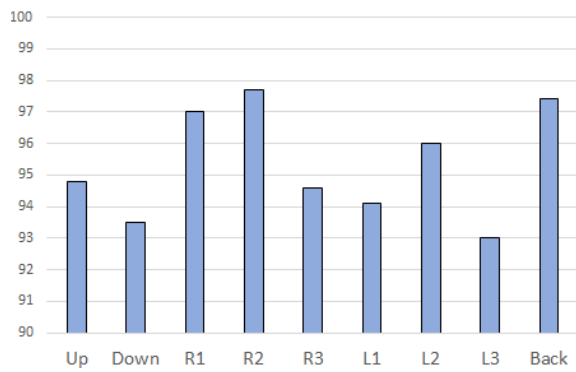

Fig. 5 Means of tapped position recognition accuracy.

bels for each tap event (e.g., R1, R2, L1, L2, and so on), we can measure the accuracy of the proposed system by testing whether it could correctly identify the positions where the tap events occurred. In our experiments, ten subjects were asked to tap 100 times for each position shown in Fig. 3(a), giving 9,000 samples in total (i.e., 10 subjects × 100 times × 9 positions = 9,000 samples). We use five-fold cross-validation. That is, we divide the 9,000 samples into five equal-sized disjoint sets and train the SVM method using four of the sets (which corresponds to 80% of the entire data) while we measure the test accuracy on the set that has not been included in the training data. We repeat this process for five times, each of which corresponds to a different test set.

Figure 5 presents the accuracy means with standard errors. As the figure indicates, the accuracy is greater than 90% for all positions, which demonstrates the feasibility of customized virtual buttons in electronic mobile devices. The accuracy of R1, R2 and Back is higher than the other positions, while that of Bottom and L3 is lower. This difference relates to the degree of accessibility of the corresponding buttons; when grasping a smartphone, it is relatively easy to tap R1, R2 and Back, but less convenient to tap Bottom and L3.

5. Related Work

Several studies have developed new input methods for electronic mobile devices using additional hardware [2]–[4]. One study [2] for example, employed a piezo-electric microphone and vibration speaker to recognize the hand shape that gripped a smartphone, while Geltouch [3] proposed a new input method using a gel-based layer. Additionally, some studies have recognized hand posture using touch sensors [4] and capacitive sensors.

A different group of studies have exploited built-in sensors in electronic mobile devices to devise new input methods [5]–[9]. Heo and Lee implemented a framework that recognizes pressure from a user's touch action on a touchscreen [5]. Another study [6] recognized hand posture and pressure from a user using inertial sensors and a touch screen, while a further study [7] exploited a tap to the bezel as a new input method. The interaction between a user and touchscreen has also been improved using pressure from a user's tap [8], and by sound profiles [9] when a user touches a touchscreen.

There are some studies that are closely related to this paper [10], [11]; the former utilizes the accelerometer to detect a tap location, and the latter does the accelerometer, gyroscope and microphones to detect richer interactions than the former. While the studies have succeeded to detect a tap location or similar events using data classification, they do not analyze principles how a single tap event changes the readings of the accelerometer and gyroscope, which can be used for intuitive understanding of feasibility of virtual buttons.

6. Conclusion and Discussions

In this letter, we demonstrated the feasibility of the design and implementation of customized virtual buttons for electronic mobile devices. Targeting nine virtual buttons on five different sides of a smartphone, we showed that the virtual buttons operate with more than 90% accuracy, regardless of user and button positions. In the future, we plan to explore how we can utilize other built-in sensors to further improve the accuracy of virtual buttons.

The biggest limitation of the current implementation is the hand-tuned thresholding, which leaves a room for automatic threshold detection. Whereas even the simple detection can lead to high accuracy, more personalized training or thresholding can be a good direction for further improvements.

Another direction of future work is to adapt the proposed virtual buttons to various situations. For example, one may wonder whether the proposed virtual buttons correctly operate when a user places a mobile device on a table; since the principle of movement and rotation by a physical touch event on a mobile device still holds, we expect the proposed virtual buttons work as long as we assign a proper threshold. Also, it is interesting to handle a situation where a user drops a mobile device onto a floor; we expect that the situation yields an abrupt change of acceleration, and hence, it is possible to distinguish the situation and the tap event on a mobile device.

Acknowledgements

A preliminary, incomplete version (2 page-long) is presented in the work-in-progress session of CPSNA 2016 [12], which does not have any formal proceedings.

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2017H1D8A2031628, 2019R1A2B5B02001794) and by the Ministry of Education (2018R1D1A1B07040321). This research was also supported by the IITP (Institute for Information & communications Technology Promotion) funded by the MSIT (Ministry of Science and ICT) (IITP-2019-2015-0-00742).

References

[1] C.C. Chang and C. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol.2, no.3, pp.1–27, 2011.
[2] M. Ono, B. Shizuki, and J. Tanaka, "Touch & activate: Adding interactivity to existing objects using active acoustic sensing," Proc. 26th annual ACM symposium on User interface software and technology, pp.31–40, 2013.
[3] V. Miruchna, R. Walter, D. Lindlbauer, M. Lehmann, R. Von Klitzing, and J. Müller, "GelTouch: Localized tactile feedback through thin, programmable gel," Proc. 28th ACM Annual ACM Symposium on User Interface Software & Technology, pp.3–10, 2015.
[4] K. Kim, W. Chang, S. Cho, J. Shim, H. Lee, J. Park, Y. Lee, and S. Kim, "Hand grip pattern recognition for mobile user interfaces," Proc. 18th Conference on Innovative Applications of Artificial Intelligence, pp.1789–1794, 2006.
[5] S. Heo and G. Lee, "Forcetap: Extending the input vocabulary of mobile touch screen by adding tap gestures," Proc. 13th International Conference on Human Computer Interaction with Mobile Devices and Services, pp.113–122, 2011.
[6] M. Goel, J.O. Wobbrock, and S.N. Patel, "Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones," Proc. 25th annual ACM symposium on User interface software and technology, pp.545–554, 2012.
[7] M. Serrano, E. Lecolinet, and Y. Guiard, "Bezel-tap gestures: quick activation of commands from sleep mode on tablets," Proc. ACM SIGCHI Conference on Human Factors in Computing Systems, pp.3027–3036, 2013.
[8] Y.C. Tung and K.G. Shin, "Expansion of human-phone interface by sensing structure-borne sound propagation," Proc. 14th Annual International Conference on Mobile Systems, Applications, and Services, pp.277–289, 2016.
[9] C. Harrison, J. Schwarz, and S.E. Hudson, "TapSense: Enhancing finger interaction on touch surfaces," Proc. 24th annual ACM symposium on User interface software and technology, pp.627–636, 2011.
[10] W. McGrath and Y. Li, "Detecting tapping motion on the side of mobile devices by probabilistically combining hand postures," Proc. 27th annual ACM symposium on User interface software and technology, pp.215–219, 2014.
[11] C. Zhang, A. Guo, D. Zhang, C. Southern, R. Arriaga, and G. Abowd, "BeyondTouch: Extending the input language with built-in sensors on commodity smartphones," Proc. 20th annual meeting of the intelligent user interfaces community, pp.67–77, 2015.
[12] S. Song, N. Kim, and J. Lee, "Towards customized virtual buttons for cyber-physical systems," the Work-in-Progress Session of the 4th IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, 2016.