Evaluating Back-of-Device Interaction for Input Using Built-in Smartphone Sensors

Harpreet Kaur Saini
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada M3J 1P3
kaurharp@my.yorku.ca

ABSTRACT

Most smartphones today have a rich set of sensors which are used to infer input; however, the primary mode of input is still confined to the front of the screen. Researchers have used different features derived from various commodity sensors (e.g., gyroscope, luminosity, etc.) to investigate the potential opportunities for interaction on the Back-of-device (BoD). For our research, we use the luminosity sensor to detect a tap on the rear camera on smartphones. This research focuses on using single and double tap for input on rear camera to perform play/pause and forward actions on the media player. The research compares the tap errors for the dominant and non-dominant hand. One of the salient findings of the experiment shows that BoD interaction is poor when using non-dominant hand and double tap error increases with the number of trials. In an evaluation with ten participants, users were able to tap with 95% accuracy.

Keywords

Smartphones; Back-of-Device interaction; Sensors; Input

INTRODUCTION

Mobile devices such as smartphones are used to perform various tasks. With the increasing capabilities of smartphones today, they provide a whole new world of user experience. Along with the increasing mobile performance, they hold vast amounts of sensitive information since they are being progressively used for e-banking, various authentication purposes, private and professional documents, and much more. Presently, the primary source of input on mobile phones is the front touchscreen.

However, when it comes to authentication tasks, the front screen interaction for input poses threats such as shoulder surfing attacks, where shoulder surfing is the practice of spying on electronic devices to obtain personal information of others [3]. There are other scenarios when front interaction on smartphones becomes inconvenient, such as when a phone is in the pocket, or when one-hand is used for input while the other hand is occupied with some secondary task, or when user's hands are dirty. To address these problems, back-of-device interaction can be used for input on smartphones using built-in sensors, such as accelerometers, gyroscopes, and microphones [4].

Back-of-device interaction has been proposed to address problems like:

- *Fat finger problem* [2, 6]: where the user's finger includes other information on touch device when selecting targets, especially on small devices. Fat finger problem is most commonly encountered when holding the device with one hand, and that hand's thumb is used for input purposes. However, the thumb of the hand holding the device is limited to reach the entire screen for input and thus gives rise to *limited thumb reachability* [7] problem. Furthermore, the wide surface area of the thumb is always prone to error when targeting a selection on a small devices.
- *Occlusion problem* [7]: when the user interact with the device, user needs to position the hand or arm above or below the screen while interacting, which might block the view of screen that might include important information.
- *Smudge attacks* [3]: where the attacker uses the latent smudges to extract the sensitive information by discerning the password pattern. So BoD has shown to be useful when it comes to privacy, by preventing the shoulder surfing attacks [5].

This feature of BoD interaction is totally dependent on the sensors. Recently, it has been shown that BoD interaction feature on smartphones can be accomplished using built-in hardware like microphones, accelerometers, or gyroscopes [5]. However, it is uncertain which derived features from sensors would allow practical use since not all sensor measurements can reliably and efficiently detect BoD interactions [4]. Furthermore, the energy consumption is a major concern in context to incorporating BoD interaction on smartphones using sensors.

Most of the research on BoD interaction assimilates the use of location information on the back of the device for touch purposes [2]. However, the input can be integrated using pressure properties where the user applies the pressure against the touchscreen for input; an existing embedded feature on Apple's iPhone 7 [2].

In this paper, we evaluate the one-handed BoD interactions for the dominant and non-dominant hand regarding tap errors. One-handed input is considered to evaluate BoD interaction because nowadays mobile communication is mostly done with one-hand only while the other hand is usually engaged with a primary or secondary task [4].

Moreover, one-handed BoD tapping interaction can be used for applications such as browsing through image, web pages; can be used for scrolling purposes; producing actions like next/previous; as well as for camera functions such as taking selfies, zooming in and out, and so forth [8].

Related Work

Back-of-device interaction has already been commercially available in the market in the form of PlayStation Vita (handheld game console), since early 2012 [3]. Other research works that have experimented to eliminate occlusion by shifting the interaction to the side or back of mobiles devices are Behind Touch, BlindSight, and LucidTouch [9].

HaptiCase, as shown in Figure 1, incorporates the added feature to the back of smartphones to use the tactile landmarks for more accurate eye-free touch input using the user sense with their fingers, covering 99% of all touches [1]. In the research, Corsten et al. [1] compared the touch performance of different landmark layouts with a regular one. They used a landmark design of dots on 3 x 5 grid, which improved the eye-free tapping accuracy and allowed the targets to be as small as 17.5mm (a reduction of 14% in target size). The study concluded that users were more accurate for eye-free indirect tapping using HaptiCase compared to having no tactile landmarks, (best: *DotsL*, OFFSET: 6.75 mm on average) [1].

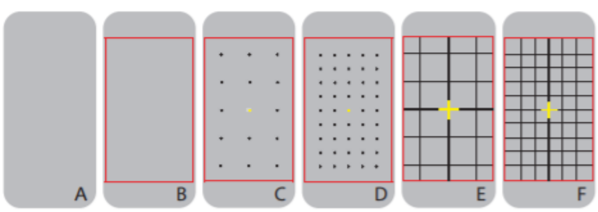


Figure 1. HaptiCase designs: A = *Base*, B = *Frame*, C = *DotsL*, D = *DotsH*, E = *LineL*. B-F provide tactile landmarks, such as the touchscreen position (red frame). Center highlighted in yellow. [1].

Luca et al. [3] conducted research which represents the BoD Shapes, an authentication method for smartphones that use back of the device for input and offers resistance to shoulder surfing. Their study compared BoD shapes to PIN authentication, Android grid unlock, and front version of the system which allowed them to compare and analyze performance, and security measures between the front and back authentication [3]. The user study was conducted on two levels of difficulty, namely, easy and hard, using given and self-selected passwords for a PIN, grid unlock, BoD front shapes and BoD back shapes, as shown in Figure 2. The results showed that a PIN performed best with only 1 basic error whereas BoD Shapes and grid unlock with hard patterns created basic error rates of 19.5% and 26.4% respectively, where basic error implies that user was able to input the password correctly on second or third try [3].

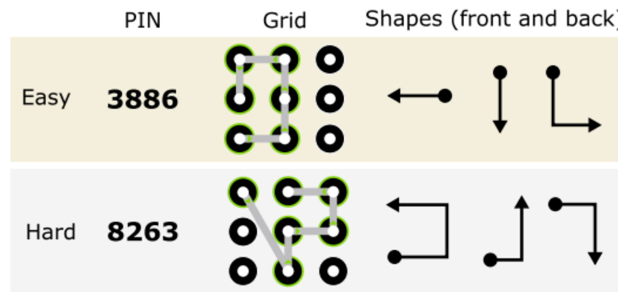
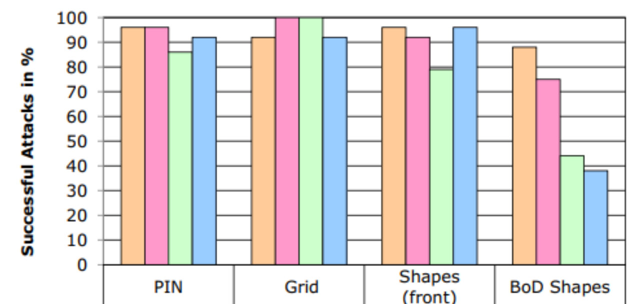


Figure 2. Examples for easy and hard PINs, patterns and shapes as used in main user study [3].

The study revealed that the authentication speed was fastest for easy self-selected passwords, where mean speed was 2.4 sec over all the systems [3]. Shoulder surfing attacks, however, showed interesting differences between the four systems as shown in Figure 3.



	PIN	Grid	Shapes (front)	BoD Shapes
Easy	96	92	96	88
Easy Self	96	100	92	75
Hard	86	100	79	44
Hard Self	92	92	96	38

Figure 3. Successful shoulder surfing attacks of four systems and the four different authentication passwords in percentage [3].

Zhang et al. [9] implemented the BackTap technique that extends the input modality to four distant tap locations: top-left, top-right, bottom-left, and bottom-right on the back of the smartphone. One of the main advantages of their tapping gestures is eye-free interaction, especially when the phone is in a user's pocket. They defined two phases for detection of a tap event: 1) segmentation where they determined that a tap has occurred; and 2) classification where they determined the location of tap on he back of the phone. The research was conducted with 11 participants over two days, with a total of 3300 taps per participant in response to randomized stimuli in four locations. Their research evaluated 92.27% to 96.67% accurate BackTap tapping interaction which depicted that over longitudinal usage, BackTap interaction can be used for a number of compelling applications. Thus, the research can be used to map the back corners of a case to volume up, volume down, track forward and previous track.

Granell and Leiva [4] focused on constructing and selecting a subset of features which will ensure low energy consumption and will best predict the BoD tapping input. They build several classifiers for single and double tap and showed that using 5-10 features helps achieve 98% of accuracy. Furthermore, interesting results were found for the BoD taps, performed using the non-dominant hand, as shown in Figure 4. High accuracy and Kappa statistics for non-dominant was high, which was explained by the fact that the user is more cautious when performing in comparison to their dominan hand.

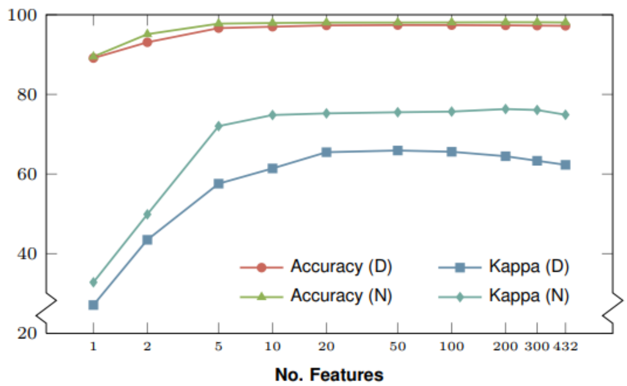


Figure 4. Results of BoD tap input detection, either single or double taps with dominant (D) and non-dominant (N) hand [4].

Granell and Leiva [5] presented βTap, a Back-of-device tap detection software, that uses commodity sensors to support single and double tap for mobile devices using an optimized feature set and a generalized linear model classifier. Some developed a set of applications using βTap are *βTap Camera*, *βTap Flutter Cow*, and *βTap Music Player*. βTap Camera application allows taking pictures by tapping at the back of the device. This application provides a single BoD tap for camera focus and allows for comfortable holding. βTap Flutter Cow, shown in Figure 5, is a game application which allows one-hand playing using BoD interaction. The objective of the game is to direct the flying cow between the set of obstacles by tapping at the back of the device. βTap Music Player application uses the single tap to play/pause the music and double tap to advance to the next song. It also facilitates play/stop casting to a nearby TV.

Figure 5. βTap Flutter Cow game [5].

Shimon et al. [7] performed an elicitation study to explore users' mapping of gestures to smartphones commands and identify their criteria for using BoD gestures. This approach presents the participants with system action and requesting them to suggest a gesture to be used as a trigger. The elicited study showed that the design gestures were influenced by *legacy bias* as some participants created 'C' and 'V' gestures for Copy and Paste, *concern for accidental input*, and *ease of performance* based on location as some participants depicted ease of use of single tap in middle of back of device to take selfie, also the use of single tap on upper left corner to open an application.

METHOD

An experiment was conducted in which participants were asked to evaluate the BoD input by using the rear camera as an input sensor to detect the tap which in-turn performs an action on the media player. The interaction then was analyzed on the basis of tap error and tap count.

Participants

Ten volunteering participants were recruited for the experiment, of the ten participants, six participants were females and four were males. Participants ranged from 20 to 30 years old. All the participants involved in the experiment were University students, all using right-hand as dominant hand and none with any motor impairments. The handedness of the participants was analyzed using Edinburg Handedness Inventory, as shown in Figure 6, which showed that all participants were right-handed.

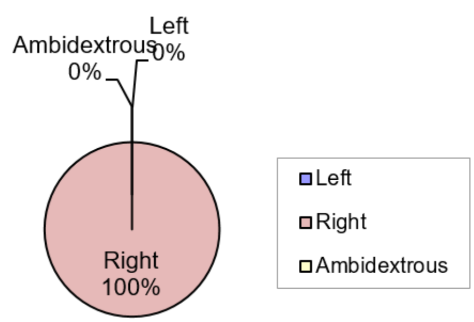


Figure 6. Edinburg Handedness Inventory result.

Apparatus

The experiment was conducted with the help of a customized Android application to study BoD interaction for input using a rear camera as the input sensor. The original code was taken from an open source [6] which was customized to perform the user study. The original code includes the implementation of both swiping and tapping on the rear camera and performs various actions on the media player. However, since both swipe and tap cannot be detected at the same time, the experiment was conducted to detect a tap on back-of-device [6]. The experiment was performed on Samsung Galaxy S7 Edge.

The android application has a media player which interacts to the tapping performed on the back-of-device. The BoD interaction is performed on the rear camera which uses the luminance computation on the ROI in an image. The application uses the rear camera to detect the finger taps and perform actions on the media player accordingly. For example, a single tap on the rear camera makes the media player play or pause, and double tap on the rear camera makes the song currently playing on the player to skip forward.

Procedure

Prior to the beginning of the experiment, each participant was asked to complete a questionnaire, detailing their demographic information, and their interaction with a mobile phone. The questionnaire included questions like:

- Hours of mobile phone usage per day.
- Do you use non-dominant hand for input on mobile phone: Yes / No
- If "yes", how often do they use their non-dominant hand for input on mobile phone?
    - Almost always
    - Often
    - Sometimes
    - Almost Never
- **Ease of use**: Level of ease with which you can perform tasks like input on phone using one hand only is:
    - For dominant hand

1	2	3	4	5	6	7
Very low						Very high

- For non-dominant hand

1	2	3	4	5	6	7
Very low						Very high

Upon completing the questionnaire, each participant was told about the purpose of the user study, which is to investigate the BoD interaction for input on the mobile devices using the rear camera.

All the participants were provided with two practice trial sessions. The practice trial was to ensure that the participants get a glimpse of what to do. For each tap setting: single and double, participants were asked to tap on the rear camera with two hand levels: dominant and non-dominant hand. For each condition, participants were asked to tap using their index finger. For each trial, the participant was asked to perform four single and four double taps alternatively on the rear camera. Thus, eight taps in total concluded one trial. At the end of each trial, completion time was computed along with total tap detected and total in-correct taps.

Design

The experiment employed a 2 x 2 x 5 within-subjects design. The factors and levels were as follows:

- Hand: *Dominant*, *Non-dominant*
- Tap: *Single*, *Double*
- Trials: *1, 2, 3, 4, 5*

Each participant was asked to perform input tasks using a single and double tap on rear camera for five trial sessions (plus one training trial) while sitting before switching between dominant and non-dominant hand. Aside from training, the amount of study was: 10 participants x 2 hand levels x 2 tap levels x 5 trials = 200 trials.

To offset learning effects, participants were divided into two groups to counterbalance the order of testing. One group performed single and double tap with dominant first, then with the non-dominant hand. The order was reversed for the other group, which performed with non-dominant hand first. The dependent variables were the Tap count and Tap error. A full description of Tap count and Tap error dependent variables are provided below.

RESULTS AND DISCUSSION

All of the trials were completed successfully. The data from the experiment was imported into a Microsoft Excel spreadsheet and the various measures were calculated, and charts were created using the calculated data.

To counterbalance the effect of learning as the participants were divided into group of two, we performed an analysis of variance to test for the possible group effect on each dependent variable. The group effect was not statistically significant ($p > .05$) for tap error and tap count. Thus, counterbalancing had the desired effect of offsetting order effect.

Tap Error

The grand mean for tap error was 4.15%. The mean tap error rate for single tap was 3.7% and for double tap was 4.6%, as seen in Figure 7.

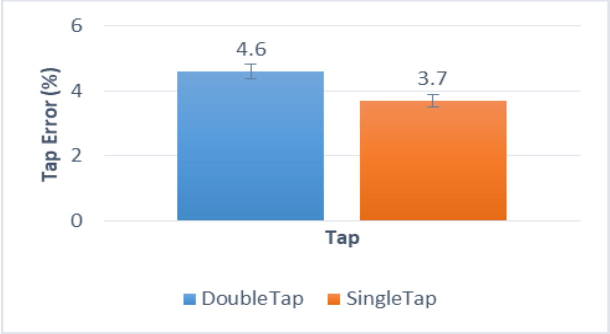


Figure 7. Mean Tap Error (%) for double tap and single tap. Error bars show ±0.05 percentage error.

The tap error rate for each participant can be seen in Figure 8, where participant P07 had the lowest tap error rate at 1.0%, while the participant P08 had the highest tap error rate at 8.0%. Among all the participants the highest tap error rate for single and double tap was 10.0% for participant P02 and P09 respectively.

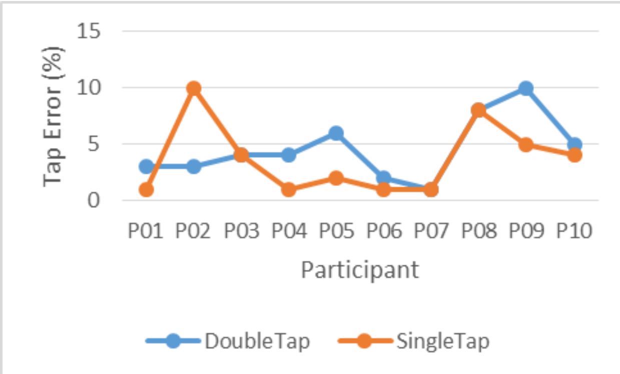


Figure 8. Tap Error (%) for all participants.

For the trial effect, the maximum tap error rate occurred during Trial 4 at 4.5% and the minimum was during Trial 2 at 3.3%, as shown in Figure 9. Over the five trials, the error rate decreased from 5.0% in the first trial to 2.9% in the second trial and increased to 3.5% in the last trial for single tap. For double tap, the error rate increased from 3.7% to 4.8% from first to the last trial. The linear relationship in Figure 9 for double tap is stated as equation of line: y = 0.361x + 3.197 and regression model is R^2 = 0.8334 and for single tap is stated as equation of line: y = -0.19x + 4.442 and regression model is R^2 = 0.1542.

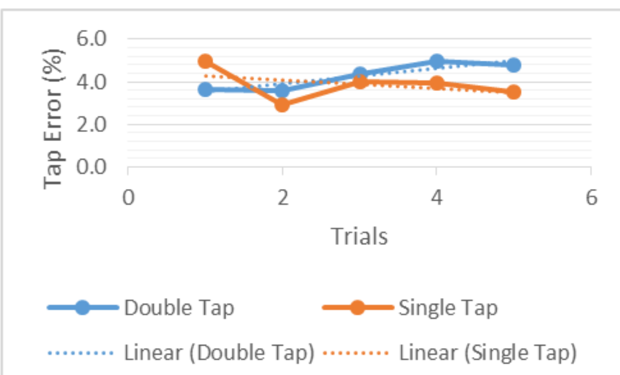


Figure 9. Results of Mean Tap error (%) for all trials.

The more detailed results are shown in Figure 10, which clearly shows the linear relationship between the tap error rates for single tap and double tap (ST & DT). The mean error rate for both single and double tap decreased from 4.3% during the first trial to 3.3% during the second trial and increased to 4.2% during the last trial, with the overall mean of 4.1%. The linear relationship in Figure 10, is stated as equation of line: y = 0.0489x + 3.905, where slope = -0.0489 and intercept = 3.905 and regression model is R^2 = 0.046.

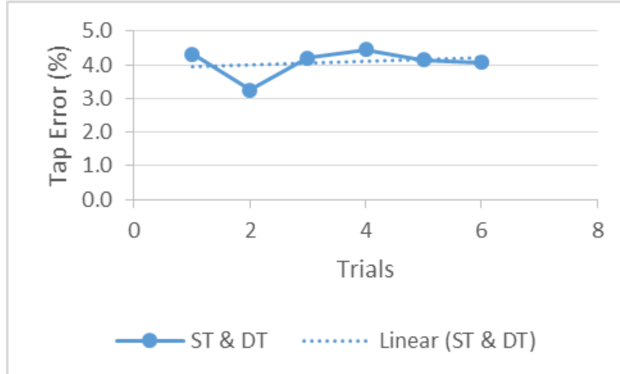


Figure 10. Mean Tap error (%) for both single and double tap (ST & DT).

Tap Count

The tap count is calculated by counting the number of single and double taps for each trial. The experiment was designed to confirm eight taps in total to conclude one trial, where participants were asked to perform four single and four double taps alternatively in each trial. Hence, the total number of single taps and double taps detected during each trial is the tap count. Obviously, the closer the number is to 4 (individually for single and double tap) for per trial, the better the detection of the tap.

The grand mean for tap count was 3.66 taps per trial. The overall mean tap count for single tap was 4.08 taps per trial, while the overall for double tap was 3.24 taps per trial. This is distinctly seen in Figure 11.

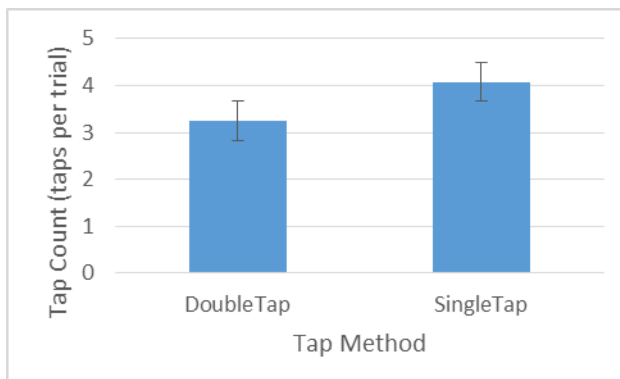


Figure 11. Mean Tap count by tap method. Error bars shows ±0.5 SE.

Among all the participants, the best performance in terms of single and double tap count was given by participant P07, mean 3.9 taps per trial for double tap and mean 4 taps per trials for a single tap. The lowest single tap count was participant P02 and lowest double tap count was for P09, a mean of 2 taps per trial, as shown in figure 12.

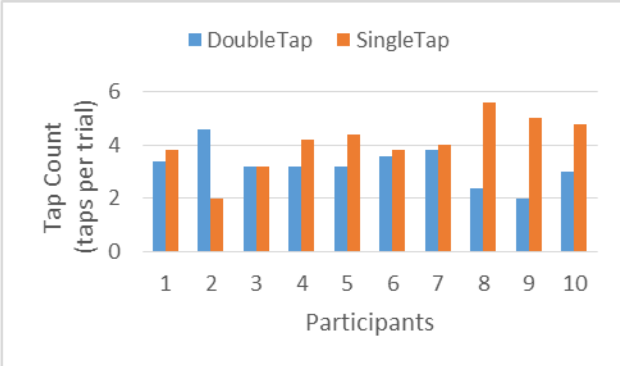


Figure 12. Mean Tap count for all participants.

The tap count over five trials is shown in Figure 13. The mean tap detection (performance) for double tap overall was fairly good, ranging from 3.0 taps per trial to 4.0 taps per trial. For single tap, however, the performance over all trials is not much stable, where it increases from 3.8 taps per trial to 5.0 taps trial from trial 1 to trial 4 and then decreases to 3.6 taps per trial for the last trial. The linear relationship in Figure 13, is stated as equation of line: y = 0.17x + 3.03 and regression model is $R^2 = 0.6123$ for double tap. For single tap, the linear equation of line: y = 0.04x + 4.004 and regression model is $R^2 = 0.0138$.

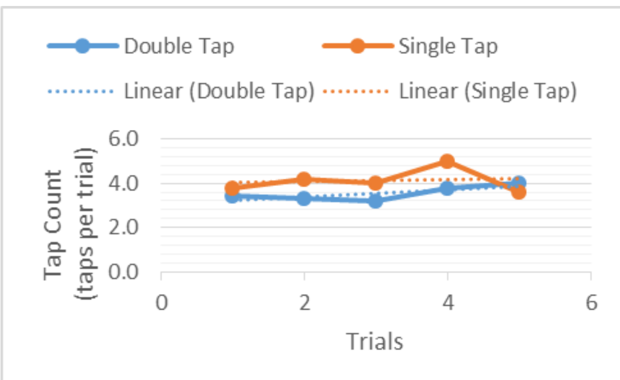


Figure 13. Mean Tap count by tap method.

The more detailed results are shown in Figure 14, which clearly shows the mean tap count for ten participants with a single tap and double tap (ST & DT) over each trial.

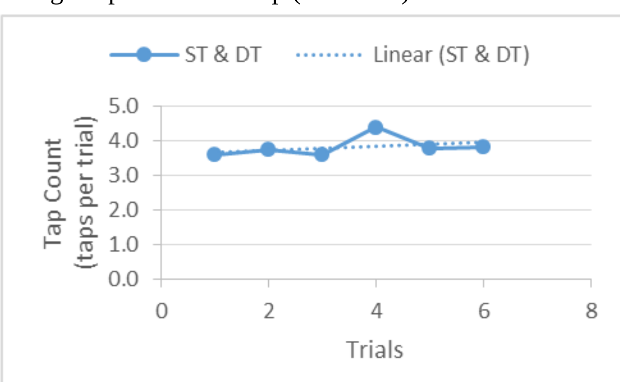


Figure 14. Mean Tap count (ST & DT)

Participant Feedback

At the end of the experiment, the participants were asked to give insight into which method (i.e., front input or back-of-device input) they prefer for input on smartphones. Most participants preferred front screen for input over BoD tapping for input. Some commented that it would be really easy to perform the tasks if the entire back-of-device is available for input. Some participants also shared their concern about the size of the phone. They expressed that the bigger phone was difficult to hold with one hand and it will be difficult to perform BoD tapping, especially with the non-dominant hand.

Most of the participants stated that they were pleased by the comfortable hand posture afforded by dominant hand as compared to non-dominant hand for BoD tapping. They expressed that it was easier to hold a phone in their dominant hand and use the index finger to tap on the rear camera. Three participants mentioned that they experienced hand fatigue due to the difficulty of holding the phone and to perform tapping on rear camera using the non-dominant hand. Two other participants commented that they like the BoD tapping feature and would love to have it incorporated with camera for taking selfies.

Overall, participants praised the single tap was more convenient for both dominant and non-dominant hands. They preferred the dominant hand for their comfortable hand posture and ease of use. When asked to rate if they would like the feature of BoD tapping on a Likert scale from 1 (least likely) to 10 (most likely), the average rating was 6.8.

CONCLUSION

We presented back-of-device interaction that lets the user perform various actions on the media player using tapping on the rear camera which uses luminosity sensor to detect the tap. In the research, we evaluated the tap error for dominant and non-dominant hand using the index finger to tap. The experiment shows that the mean error rate for a single tap is 3.7% and for the double tap is 4.6%. Thus, the accuracy lies from 95% to 96% to correctly identify tap.

The accuracy can be improved with longitudinal learning. We also observed that the tap errors were influenced by hand fatigue for the non-dominant hand. Since the participants depicted that they had experienced fatigue when they were trying to perform tap using an index finger while using their non-dominant hand.

The understanding of the BoD interaction using the built-in sensors like luminosity, accelerometer, etc. can help to incorporate new features into the smartphones which enables us to perform certain tasks with great ease. Overall, BoD interaction can be used either as a means of direct input or as a complementary input aid.

REFERENCES

1. Corsten, C., Cherek, C., Karrer, T., & Borchers, J. (2015). HaptiCase: Back-of-device tactile landmarks for eyes-free

absolute indirect touch. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15,* (New York: ACM, 2015), 2171-2180.

2. Corsten, C., Daehlmann, B., Voelker, S., & Borchers, J. (2017). BackXPress: Using back-of-device finger pressure to augment touchscreen input on smartphones. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17,* (New York: ACM, 2017), 4654-4666.
3. De Luca, A., Von Zezschwitz, E., Nguyen, N. D., Maurer, M., Rubegni, E., Scipioni, M. P., & Langheinrich, M. (2013). Back-of-device authentication on smartphones. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, (New York: ACM, 2013), 2389-2398.
4. Granell, E., & Leiva, L. A. (2016). Less is more: Efficient back-of-device tap input detection using built-in smartphone sensors. *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces - ISS '16*, (New York: ACM, 2016), 5-11.
5. Granell, E., & Leiva, L. A. (2017). βTap: Back-of-device tap input with built-in sensors. *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '17,* (New York: ACM, 2017), 1-6.
6. Marques (2018). Back-of-device. *GitHub repository,* https://github.com/everettmthunzi/back-of-device, (accessed 15 March 2019).
7. Shimon, S. S., Morrison-Smith, S., John, N., Fahimi, G., & Ruiz, J. (2015). Exploring user-defined back-of-device gestures for mobile devices. *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '15*, (New York: ACM, 2015), 227-232.
8. Zhang, C., Guo, A., Zhang, D., Li, Y., Southern, C., Arriaga, R. I., & Abowd, G. D. (2016). Beyond the touchscreen: An exploration of extending interactions on commodity smartphones. *ACM Transactions on Interactive Intelligent Systems*, *6*(2), 1-23.
9. Zhang, C., Parnami, A., Southern, C., Thomaz, E., Reyes, G., Arriaga, R., & Abowd, G. D. (2013). BackTap: Robust four-point tapping on the back of an off-the-shelf smartphone. *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology - UIST '13 Adjunct*, (New York: ACM, 2013), 111-112.