Detecting Tapping Motion on the Side of Mobile Devices By Probabilistically Combining Hand Postures

William McGrath*
Stanford University
wmcgrath@stanford.edu

Yang Li
Google Research
yangli@acm.org

ABSTRACT

We contribute a novel method for detecting finger taps on the different sides of a smartphone, using the built-in motion sensors of the device. In particular, we discuss new features and algorithms that infer side taps by probabilistically combining estimates of tap location and the hand pose—the hand holding the device. Based on a dataset collected from 9 participants, our method achieved 97.3% precision and 98.4% recall on tap event detection against ambient motion. For detecting single-tap locations, our method outperformed an approach that uses inferred hand postures deterministically by 3% and an approach that does not use hand posture inference by 17%. For inferring the location of two consecutive side taps from the same direction, our method outperformed the two baseline approaches by 6% and 17% respectively. We discuss our insights into designing the detection algorithm and the implication on side tap-based interaction behaviors.

Author Keywords

Mobile interaction; motion gestures; machine learning.

INTRODUCTION

The capacitive touchscreen, as the major input medium of mobile devices, has been overloaded with interaction behaviors. To access rich functionalities and data items on a mobile device, a user often needs to navigate deep hierarchies of interfaces and deal with mode-switching issues, which can be frustrating, time consuming and error-prone. Thus, it is important to enable new input events to expand the bandwidth of mobile interaction.

Previous work has explored a variety of techniques for enhancing existing touchscreen behaviors (e.g., [1, 2]) or expanding the input vocabulary of mobile devices (e.g., [5, 9, 11]). In particular, side taps—tapping on the side of the physical frame of a mobile device—has shown promise. As a natural extension of tapping actions beyond the touchscreen, side taps can be performed eyes-free and do not cause the user to block the screen with their fingers.

Previously, Ronkainen et al. investigated the feasibility and usability of tap input using custom sensors and platforms [8], which revealed the usefulness of side taps and the challenges for detecting them reliably. Recently, Samsung Galaxy S III smartphones allowed a user to double tap the top of the phone as a shortcut for moving to the top of a list [10]. We intend to develop methods to not only detect the occurrence of a side tap event but also the side of the phone that is tapped, which provides additional parameters for activating tap location-specific functionality (see Figure 1).

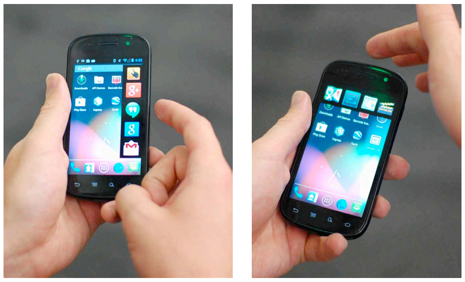

Figure 1. The user taps the right and the top side of a phone that bring out different sets of shortcuts.

In this paper, we devised a novel method for detecting finger-tapping motion on the sides of a smartphone using the built-in inertial sensors of the phone, with no additional hardware. To predict which side of the device the user has tapped, our method uses the hand pose—which hand is holding the device—as a hidden variable and then probabilistically combines estimates of the hand pose and tap location, which effectively reduces the complexity for classification and improves the accuracy compared to previous methods. This approach can potentially be applied to other handheld motion inference situations. We elaborate on the design of our detection method and its performance in comparison to other baseline methods. We also discuss its implication in designing side tap-based interaction.

RELATED WORK

Previous work has explored a variety of enhancements for mobile devices for detecting new interaction events, such as using acoustic sensors [5], custom capacitive touchscreens [1, 11], pressure sensors [13], or even building custom smartphones [8]. Although these methods demonstrated

UIST '14, October 05 - 08 2014, Honolulu, HI, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3069-5/14/10…$15.00.
http://dx.doi.org/10.1145/2642918.2647363

* The work was done during an internship with Google Research.

potential for detecting new events, they rely on custom sensors or platforms, which limits their practicality.

In contrast, on-board sensors such as accelerometers and gyroscopes are widely available on commodity devices. However, these sensors are often less expressive or precise than custom ones for capturing specific behaviors. As a result, prior work has mostly employed on-board sensors for detecting easily distinguishable motion events [9, 10] or used them as auxiliary channels for inferring or disambiguating touchscreen input [2, 6, 7, 12]. For example, previous work investigated the feasibility of inferring tap location on the touchscreen, using motion sensors, as the user types on a soft keyboard on the touchscreen [6, 7]. Prior work also developed methods for improving touchscreen text entry by compensating for extraneous movement due to the user walking [2], or dynamically adapting for detected hand postures [3, 4].

Compared to the prior art, we focused on detecting tap events on the side of a device. Although early work has investigated the usefulness of side taps [8], little work in the literature has discussed the methods for detecting the occurrence and location of these events, especially using on-board sensors on commodity mobile devices. BezelTap combines a tap on the tablet bezel and a successive tap on the touchscreen for fast activation [12], rather using the tap on the bezel as a standalone event. Spelmezan et al. augmented an iPod touch device with continuous pressure sensors on its side for easy navigation [13]. Ronkainen et al. investigated the usability of tap input using custom sensors and platforms [8] and revealed the challenges for detecting them reliably. Recently, Samsung Galaxy S III smartphones enabled double tapping on the top of the phone as a shortcut event [10]. However, none of the previous work provides a viable solution for detecting side tap events (especially single taps) and tap locations using built-in mobile sensors. In addition, the previous work lacks technical details and empirical analyses, which makes it difficult for comparison and improvements for further research.

Because mobile devices are often handheld while used, it is natural to consider the hand posture in detecting motion. Prior work has demonstrated the usefulness of using hand postures to improve interaction and inference [1, 3, 4]. Compared to previous work, our method probabilistically combines the estimate of the hand posture and tap location, rather than deterministically using the best guess of the holding hand for further inference as the prior work does. Our method is theoretically sound and it outperformed the deterministic use of hand postures.

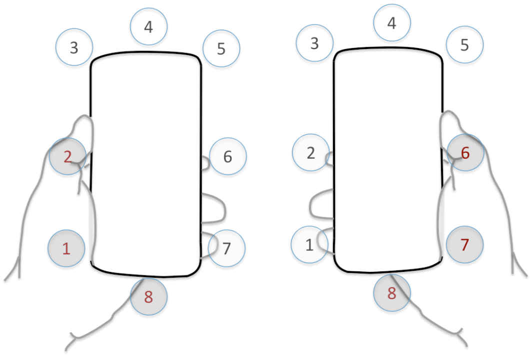

Figure 2. The side tap targets that we sampled for each holding hand during data collection. The locations labeled red in shaded circles are excluded for that holding hand.

THE DESIGN OF SIDE TAP DETECTION METHODS

To scope our exploration, we focus on detecting side taps when the mobile device is in use, rather than enabling an always-active motion gesture such as what PocketTouch does [11]. We also aim at detecting which side of the device is being tapped to allow location-specific behaviors.

Data Collection

To reliably detect side taps and their locations, we collected side tap samples—positive samples—and several hours of ambient data—negative samples—from 10 participants (6 female, 4 male). They ranged in age from 20 to 32 years old. 9 were right-handed and one was left-handed. All but one were regular smartphone users.

Procedures & Tasks. Participants were asked to complete tasks in two situations: *seated* versus *walking*. In the seated condition, participants were seated comfortably in an office chair. In the walking situation, participants were asked to walk at a natural pace following a square path roughly 20 feet on a side. We focus on the posture where the user is holding the smartphone with one hand and interacting with the device using their other hand. We asked participants to alternate the holding and interaction hands during the study.

To collect the motion data produced during everyday smartphone usage, we asked participants to perform a list of typical smartphone tasks, such as writing messages and browsing the Internet. We also asked participants to press hard buttons on the side of the device such as the volume buttons since they might generate similar motion to side taps. We then asked participants to execute a series of side taps on the device. Although we ultimately only classify a tap as top, left, or right, we instructed the participants to tap the side of the phone at five of the eight compass directions with the index finger of their hand that was not holding the device (see Figure 2), a common interaction scenario. The order that the directions were presented was randomized. Three locations, marked as red labels in the shaded circles, for either hand position were excluded because they were either awkward to reach or often blocked by the holding hand. Since each participant was instructed to perform 5 sets of 10 taps under each condition (2 use situations x 2 holding hands), the process collected 2000 side tap samples. However, issues during one trial reduced the useful number of samples to 1800 across 9 users.

Data Processing. Our data collection was conducted using a Samsung Galaxy S phone running Android. Our data collection tool instructed the participants to hold the device with a specific hand and tap on a target location in a 2-

second window. A side tap typically lasts from 100ms to 300ms, but a 2-second window allows the participant time to react to the tool's prompt. Our tool recorded the phone's accelerometer and gyroscope data as it was used. Ambient data was collected continuously whenever the tool was not expecting a tap, i.e., outside of the tap window.

Because using the entire 2-second window of a sample can introduce additional noise and might incorrectly capture consecutive side taps at runtime, we extracted a 300ms positive sample containing the largest absolute acceleration value from each original tap sample based on the observation that a side tap tends to co-occur with the largest absolute acceleration value. For the ambient motion data, we extracted negative samples by segmenting the ambient motion stream of each participant using a 300ms-sliding window, which generated 26000 negative samples in total.

Featurization

To effectively detect side taps, we need to design features that can characterize side tap motion. In particular, we hypothesize that a side tap motion consists of two physical stages: the *impact* motion that is generated when a side tap impacts the device towards a specific direction and then the opposite *return* motion that is when the user's arm brings the device back to its original position after the impact. Based on the collected side tap samples, we found that some side tap samples are consistent with our hypothesis (see Figure 3). We can easily calculate the impact direction of a side tap using the magnitude of acceleration on the X and Y axes at the first spike—*atan2*(y, x).

However, we found many other side tap samples missed either the impact or the return acceleration spike. One important reason for this phenomenon is that a side tap happens too fast for the accelerometers in our experimental smartphone (sampling at 50HZ) to reliably capture these spikes. To make our approach robust to low acceleration sampling rates, we only attempt to capture one spike in a side tap and calculate a direction angle feature based on the acceleration along the sensor's X and Y axes when either records the largest absolute magnitude of acceleration during the tap. However, without knowing if the spike is from the impact or the return motion, we cannot reliably determine which side the tap comes from. Instead, the direction angle feature is a good discriminator for determining the holding hand (see Figure 4). Since the holding hand serves as a pivot, side tapping causes the device moves diagonally rather than along perfect horizontal or vertical directions.

In addition, to capture the characteristics of a side tap that are less obvious, we added brute force features that are frequently used by previous work for motion detection, such as the mean, the standard deviation, skewness and kurtosis of each sensor axis. We also implemented correlation features between the axes of the same sensor, such as the 1-norm, Infinity norm and Frobenius norm by treating each axis as a matrix column [6], and between sensors, such as Pearson correlation coefficients. For the following training and prediction stages, the feature values of each sample are standardized based on the mean and variance of each feature of training samples. We also normalized each feature vector with its L2 norm.

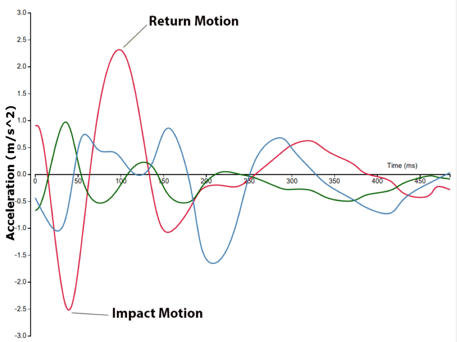

Figure 3. The interpolated acceleration plot of a side tap sample on the X (red), Y (green) and Z (blue) axes when holding the phone with the left hand and tapping the right side of the phone with the index finger of the right hand.

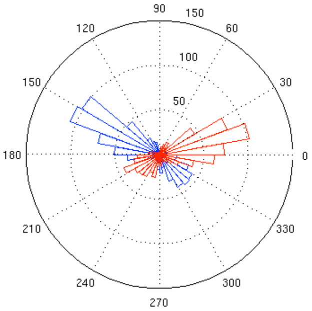

Figure 4. An angle histogram of side taps from all sampled directions while holding the smartphone with the left hand (blue) vs. the right hand (red).

Recognizing Side Taps

There are two steps in recognizing side taps. We first need to detect the occurrence of a side tap event from the continuous sensor streams. Once detected, we need to determine which side of the device the user has tapped.

Detecting Side Tap Occurrences. We trained a Linear SVM classifier based on the side tap data (the positive samples) and the ambient data (the negative samples). Because this classifier needs to continuously run, we decided not to use the matrix features for detection to save computation, as the results were acceptable without their inclusion. Because the ambient data has significantly more samples, we balanced the training data set of positive and negative samples to avoid negatively skewing our classification model. Based on a 9-fold cross validation splitting on participants—8 participants' data for training and 1 participant's for testing, our tap detection classifier achieves high accuracy 97.9% with mean precision of 97.3% and recall 98.4%. In particular, we found side taps were rarely confused with physical side-button presses (such as the volume buttons), because side taps are generally more swift and forceful, which generate an acceleration of much larger magnitude in a short period of time than button presses.

Inferring Side Tap Locations. We infer which side of the device the user has tapped, $Side_{tap}$, as the side, L, that has the highest probability based on sensor readings, s.

$$Side_{tap} = \underset{L\in\{Left,Top,Right\}}{\operatorname{argmax}} P\left(L|s\right) \qquad (1)$$

As mentioned in the previous section, there are three target sides, i.e., the left, top and right side. The bottom side is excluded because it is ergonomically difficult to use. Previous work has shown that knowing the hand posture can improve the efficiency of follow-up predictions [3]. This insight is important in our context—there are only two of the three sides usable in each holding hand condition. As a result, we decide to leverage the holding hand information in our tap location inference by introducing a hidden variable H (see Equation 2).

$$P\left(L|s\right) = \sum_{H\in\{Left,Right\}} P\left(L|Hs\right)P\left(H|s\right) \qquad (2)$$

H represents the hand holding the device—left or right. Notice that in contrast to prior work that uses the best guess of the holding hand deterministically and switches its follow-up prediction model, we probabilistically combine the outcome of the holding hand prediction and the tap location recognition, conditioned on a given holding hand. We hypothesize that our probabilistic approach could outperform the deterministic approach due to imperfect holding hand detection.

Equation 2 uses two types of classifiers: one infers the holding hand and the other infers tap locations given each holding hand—that includes two separate classifiers, one for each holding hand. We trained the holding hand binary Linear SVM classifier based on the tap samples. It operates on the full feature vector that includes the matrix features, since it will only be invoked once a tap is detected. Based on a 9-fold cross validation on participants, the classifier achieves 89% mean accuracy (SD=6%), which is encouraging but not perfect, as we speculated.

We then trained a tap location classifier (also using a Linear SVM) for each of the holding hand conditions. Because our side tap samples were collected for specific locations on the side, there are several ways to train the classifier. We can train a binary classifier for the two target sides in each holding hand condition, by merging the data belonging to each side, e.g., location 6 and 7 belong to the right side and location 3 and 4 belong to the top (see Figure 2). The corner locations such as location 5 can be considered as a part of either side, so their data can be left out or shared by both the sides. Alternatively, we can train a multi-class classifier that infers 5 locations in each holding hand condition and then combine their probabilities according to the side to which they belong. Empirically we found the second strategy that combines probabilities outperforms the first approach that merges the training data.

One important detail in calculating Equation 2 is that we need to transform SVMs scores to probabilities before they can be combined. This is important to make the impact of each component in Equation 2 probabilistically proportional on the final outcome. We achieved this by fitting a probability density function (using a Gaussian mixture model) over the output scores of each classifier in Eq. 2 based on the training data.

To find out how our approach performs, we compared it with two baseline approaches: directly predicting the tapped side without using holding hand information or using the holding hand information deterministically—that resembles the previous work [3]. We used Linear SVMs based on the same dataset for these baseline methods. Based on a 9-fold cross validation, our probabilistic combination approach achieved 83% mean accuracy (SD=8%), which outperforms the two baseline approaches: 80% (SD=8%) with deterministic use of holding hand information and 66% (SD=9%) with no holding hand used (see Figure 5), χ^2_2 =102.48, p<.01. The difference between the Probabilistic and Deterministic methods is not significant, χ^2_1 =3.04, p=0.08, due to the high variance across participants. However, after we remove the two lowest performers (participant 6 and 7) from analysis, the difference between the two methods becomes significant, χ^2_1 =4.1025, p=0.04.

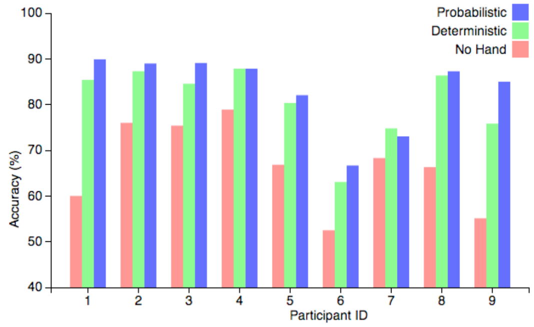

Figure 5. The accuracy for inferring the location of single side taps for each round in a 9-fold cross validation, under three conditions: probabilistically combining hand postures (Blue), deterministically using the best guess of hand postures (Green), and directly inferring tapping location without hand posture inference (Red).

Using single side taps as a building block, we can construct a variety of compound interaction events by combining side taps from the same or different locations. We analyzed how well our method can detect the location of two consecutive single taps from the same direction using the same feature set, based on data samples synthesized from our single tap data. We found the accuracy of our method achieved 90% (SD=8%) accuracy, which again outperformed the baseline methods: 84% (SD=10%) of the deterministic use of hand postures and 73% (SD=12%) when no hand posture inference is involved (see Figure 6), χ^2_2 =127.32, p<0.01. The pairwise difference between Probabilistic and Deterministic methods is significant, χ^2_1 =13.32, p<0.001, even when all the participants are included. This analysis indicated that higher accuracy can be acquired for detecting the locations of a side tap sequence.

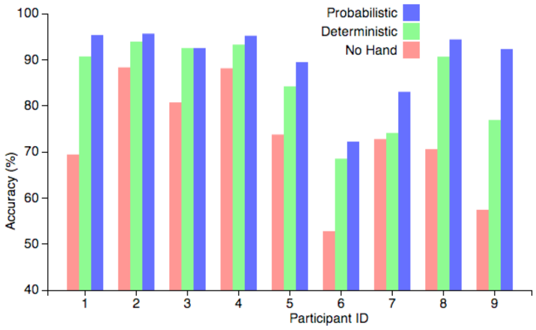

Figure 6. The accuracy for inferring the location of two consecutive single taps from the same side of the device.

DISCUSSIONS

The experiments showed that our method can reliably detect the occurrences of single side tap events using smartphones' built-in motion sensors. This presents substantial progress beyond prior work in the literature [8] as well as existing commercial products [10] that only detect double side tap occurrences. In addition, while prior work only discerns one tap direction, our method infers more details about a side tap event—which of the three sides of the device the user has tapped—to enable invoking tap location-specific actions (see Figure 1).

Our method demonstrated an accuracy advantage over two baseline methods in detecting side tap locations. But, the mean accuracy of our method for inferring single tap locations is still less than ideal for a fundamental interaction event. Because our method is data-driven, its performance is largely determined by the quality of training data we collected. Excluding low quality data, such as Participant 6's, from both training and testing can tremendously improve the reported accuracy. In addition, the motion sensors of our data collection phone are less sensitive than latest smartphones. Using a motion-sensitive device is crucial as a single tap event occurs extremely fast (<300ms). With the latest models, we expect better accuracy from our method.

Our method has only been trained and tested in a two-handed interaction scenario. There does not seem to be a fundamental reason that our method should have trouble with other scenarios such as tapping with the holding hand provided that it has been trained with sufficient data from those scenarios. However, there could be subtleties in feature analysis for these scenarios, e.g., we might see more motion along the Z-axis in the one-handed situation.

Lastly, an important consideration in designing side tap interaction is how to surface feedback to the user about side tap's activation status. For instance, a weak single tap could show a visual indication that certain functionality is associated with a side of the device, without activating it. A stronger single tap could skip the preview stage and activate the functionality immediately. We can easily enable this interaction scenario using a dual threshold mechanism, i.e., a lower threshold for tap detection for weak taps and a higher threshold for stronger taps—a higher precision but a lower recall.

CONCLUSIONS

We contributed a novel method for detecting side taps and their location as input events for button-less mobile interaction. The experiments indicated that our method can reliably detect side tap occurrences and outperformed two baseline approaches in detecting tap locations. Our concept of factoring hand postures probabilistically into further motion inference can be generalized and potentially benefit other handheld motion inference problems.

REFERENCES

1. Cheng, L.-P., Liang, H.-S., Wu, C.-Y., and Chen, M.Y. iGrasp: grasp-based adaptive keyboard for mobile devices. CHI'13. 3037-3046.
2. Goel, M., Findlater, L., and Wobbrock, J. WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. CHI'12. 2687-2696.
3. Goel, M., Jansen, A., Mandel, T., Patel, S.N., and Wobbrock, J.O. ContextType: using hand posture information to improve mobile touch screen text entry. CHI'13. 2795-2798.
4. Goel, M., Wobbrock, J., and Patel, S. GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. UIST'12. 545-554.
5. Harrison, C., Schwarz, J., and Hudson, S.E. TapSense: enhancing finger interaction on touch surfaces. UIST'11. 627-636.
6. Miluzzo, E., Varshavsky, A., Balakrishnan, S., and Choudhury, R.R. Tapprints: your finger taps have fingerprints. MobiSys'12. 323-336.
7. Owusu, E., Han, J., Das, S., Perrig, A., and Zhang, J. ACCessory: password inference using accelerometers on smartphones. HotMobile'12. 1-6.
8. Ronkainen, S., Häkkilä, J., Kaleva, S., Colley, A., and Linjama, J. Tap input as an embedded interaction method for mobile devices. TEI'07. 263-270.
9. Ruiz, J. and Li, Y. DoubleFlip: A Motion Gesture Delimiter for Mobile Interaction. CHI'11. 2717-2720.
10. Samsung. *Galaxy SIII,* http://www.samsung.com/us/support/owners/product/SPH-L710RWPSPR.
11. Saponas, T.S., Harrison, C., and Benko, H. PocketTouch: through-fabric capacitive touch input. UIST'11. 303-308.
12. Serrano, M., Lecolinet, E., and Guiard, Y. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. CHI'13. 3027-3036.
13. Spelmezan, D., Appert, C., Chapuis, O., and Pietriga, E. Side pressure for bidirectional navigation on small devices. MobileHCI'13. 11-20.