Demonstrating PalmTouch: The Palm as An Additional Input Modality on Commodity Smartphones

Huy Viet Le
University of Stuttgart
Stuttgart, Germany
huy.le@vis.uni-stuttgart.de

Sven Mayer
University of Stuttgart
Stuttgart, Germany
sven.mayer@vis.uni-stuttgart.de

Thomas Kosch
LMU Munich
Munich, Germany
thomas.kosch@ifi.lmu.de

Niels Henze
University of Regensburg
Regensburg, Germany
niels.henze@ur.de

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
MobileHCI '18 Adjunct, September 03–06, 2018, Barcelona, Spain.
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5941-2/18/09.
https://doi.org/10.1145/3236112.3236163

Abstract
Touchscreens are the most successful input method for smartphones. Despite their flexibility, touch input is limited to the location of taps and gestures. We present *PalmTouch*, an additional input modality that differentiates between touches of fingers and the palm. Touching the display with the palm can be a natural gesture since moving the thumb towards the device's top edge implicitly places the palm on the touchscreen. We developed a model that differentiates between finger and palm touch with an accuracy of $99.53\,\%$ in realistic scenarios. In this demonstration, we exhibit different use cases for *PalmTouch*, including the use as a shortcut and for improving reachability. In a previous evaluation, we showed that participants perceive the input modality as intuitive and natural to perform. Moreover, they appreciate *PalmTouch* as an easy and fast solution to address the reachability issue during one-handed smartphone interaction compared to thumb stretching or grip changes.

Author Keywords
Palm; capacitive image; machine learning; smartphone.

ACM Classification Keywords
H.5.2 [User Interfaces]: Input devices and strategies

Figure 1: *PalmTouch* during one-handed use.

Figure 2: *PalmTouch* with the flat hand during two-handed use.

Figure 3: *PalmTouch* by forming a fist during two-handed use.

Introduction and Background

Smartphones have recently become the most successful mobile devices. Through a touchscreen, smartphones offer a wide range of functions that are used by millions of people. While most functions are accessible within a number of touches, some are used so frequently that shortcuts were introduced. Traditional devices offer volume buttons and a power button to enable users to change the device's volume and state with just one press. With an increasing number of frequently used functions such as device assistants, cameras or music players, device manufacturers and developers look for new ways to integrate them for faster access. Recently, Samsung introduced a dedicated hardware button to call the Bixby assistant on the Samsung Galaxy S8. The HTC U11 incorporates Edge Sense, a pressure sensitive frame that launches a user-defined action when squeezing the device. While these additional input controls fulfill their purpose, they require additional hardware which leaves out the possibility to update already existing and older devices. Further, these solutions clutter the device itself and become inconvenient when users are not holding the device in their hand or are using a bumper case.

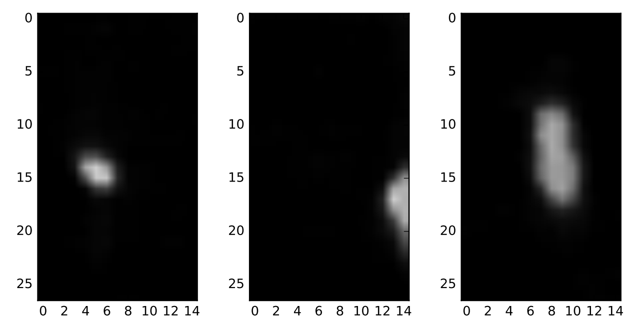
Touch gestures constitute a possible solution to the challenge described above. For example, OnePlus devices recognize gestures on the lock screen to launch user-defined applications. However, system-wide drawn gestures that are always accessible for user-defined actions may conflict with other applications. Previous work presented a number of alternative input modalities to support traditional multi-touch input. This includes using the finger's 3D orientation [12, 13], contact size [1], or pressure [5]. While these enrich the information of a finger's touch, they also bring restrictions since specific finger postures may now trigger unwanted actions. One solution to lower the likelihood of triggering unwanted actions is to differentiate between fingers or parts of fingers, which prevents interference with the main finger for interaction. Previous work [2, 3, 4] differentiated between different parts of the finger (*e.g.* knuckle) or fingers themselves to assign unique touch actions.

Motivated by previous work, we applied this concept for one-handed as well as two-handed smartphone interaction. As a result, we present *PalmTouch*, an additional input modality that enables people to use the palm to trigger pre-defined functions instead of simply rejecting palm input as recent smartphones do. We show that this is a natural and fast gesture especially when the device is held one-handed. Stretching the thumb towards the top edge to access targets that are out of reach often places the palm on the touchscreen implicitly and subtly as shown in Figure 1. The placement is often caused by unawareness of users which suggests that this gesture can be performed naturally. Figures 1 to 3 shows three examples of using *PalmTouch* in one-handed and two-handed scenarios to trigger assigned functions.

In this demonstration, we exhibit *PalmTouch* together with use cases that we presented and evaluated in our paper [7]. We developed a model that reliably differentiates touches from fingers and palms with an accuracy of $99.57\,\%$. The use cases that we will demonstrate include the use of *PalmTouch* as a shortcut to launch a pie menu (or pre-defined applications), and to improve reachability during one-handed use. We will further exhibit *PalmTouch* with an application that shows the raw data and how the identification and classification look like. Further, we will also show how *PalmTouch* can be used to perform gestures to *e.g.*, switch or close applications. As described in the paper, *PalmTouch* works on off-the-shelf smartphones.

PalmTouch and Implementation

PalmTouch is an additional input modality for a wide range of functions. We applied the idea of hand part specific touch interaction presented in previous work (*e.g.*, using different fingers [2, 3] or finger parts [4]) for one-handed as well as two-handed interaction scenarios. Since using other fingers than the thumb or other parts of the hand (such as a knuckle) can be inconvenient or even infeasible during one-handed interaction, we instead use the palm for interaction. During one-handed interaction, the palm can be placed subtly on the touchscreen by moving the thumb towards the upper edge of the device while stabilizing the device with fingers on the left edge as shown in Figure 1. During two-handed interaction, *PalmTouch* can be used by placing the flat hand (see Figure 2) or by forming a fist on the touchscreen (see Figure 3).



(a) Finger Touch **(b)** Palm (Fist) **(c)** Palm (Flat)

Figure 4: Exemplary raw capacitive images from the data collection study. Figure (a) shows the finger of participant 5 during the dragging task; (b) shows the palm of participant 19 in the one-handed condition and (c) shows the palm of participant 9 in the two-handed condition.

We implemented *PalmTouch* using the capacitive images on an LG Nexus 5 (c.f. [9]), and the machine learning model as described in the original paper [7] to classify touches. An exemplary sample of the capacitance data on an LG Nexus 5 is shown in Figure 4. After experimenting with basic machine learning models known from previous HCI work (e.g., Support Vector Machine (SVM), Random Forest (RF)), we found that a Convolutional Neural Network (CNN) performs the best with an accuracy of $99.53\,\%$ in an validation with realistic scenarios. We used *TensorFlow* to train the model, and *TensorFlow Mobile* to run the CNN on a LG Nexus 5. Hereby, classifying one capacitive image including the blob detection takes $7.5\,ms$ on average (min = $4\,ms$, max = $11\,ms$, SD = $1.6\,ms$) over 1000 runs.

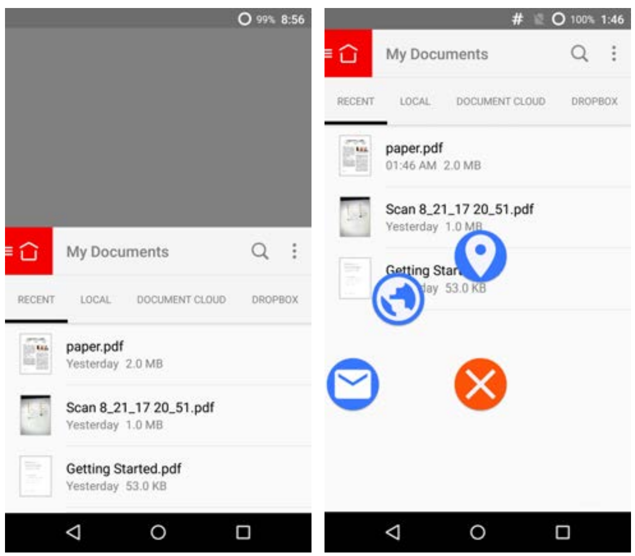
Use Cases for Demonstration

In this demonstration, we show three use cases that attendees can test on an LG Nexus 5. A video demo can be seen online[1].

Improving Reachability during One-Handed Interaction
Large smartphones pose challenges in reachability since they require changing the hand grip when used one-handed [10, 11]. With *PalmTouch*, users can stretch the thumb towards the top as if they would tap the target. This action implicitly places the palm on the touchscreen and can be used by *PalmTouch* to shift down the screen by half its size (cf. Figure 5a and [6, 8]). In our demonstration, the user can open the notification center by placing the palm on the touchscreen instead of dragging down the notification bar manually. Especially on large smartphones, this saves a grip change that could potentially lead to dropping the device.

[1] www.youtube.com/watch?v=dAo3uYnZywA

Custom Actions and Applications
Smartphone manufacturers recently integrated simple and binary input modalities such as an extra button (Bixby button on the Samsung Galaxy S8) or a squeeze on the device's edge (Edge Sense on the HTC U11) to launch pre-defined applications. While these features require additional hardware, *PalmTouch* can be readily deployed onto recent and older off-the-shelf smartphones, *e.g.*, through software updates. We demonstrate this use case by letting users open a pie menu as shown in Figure 5b. This menu enables the user to open a set of pre-defined applications. Since



(a) Reachability **(b)** Pie Menu

Figure 5: Figure (a) shows how *PalmTouch* improves reachability for one-handed use and Figure (b) show *PalmTouch* as a shortcut to launch pre-defined applications.

PalmTouch can be used eyes-free similar to a hardware button or squeeze, actions such turning off the screen or accepting a call could also be mapped to a palm touch.

Additional Input Dimensions
In addition to a binary action, *PalmTouch* offers further dimensions that can be used for interaction. The contact area's centroid can be used as a proxy for the palm touch location. This enables us to implement directional gestures, such as swiping up with the opposite hand's palm to exit an app and swiping left or right to switch between apps. The location of the opposite hand's palm can also be used to map functions to different locations of the touchscreen. For example, a palm touching the top half of the display skips to the next music title while a touch on the lower half plays the previous title. The location can also be used for a same-side palm touch (*e.g.*, x-position describes the used hand) to launch different actions depending on the used hand.

Conclusion

In this demonstration, we exhibit *PalmTouch*, an additional input modality on commodity smartphones that differentiates between touch input made by fingers and palms. We demonstrate the capability of *PalmTouch* using different use cases. This includes improving reachability, providing an additional shortcut and a technical demonstration of further input dimensions, such as directional gestures using the palm. We evaluated *PalmTouch* as well as the use cases (described in [7]) and found that users find them intuitive and natural to use. As *PalmTouch* be readily deployed into recent smartphones using software updates, *PalmTouch* provides can also compensate the removed home button on devices with an edge-to-edge display (e.g., iPhone X).

Acknowledgements
This work is supported through project C04 of SFB/Transregio 161, the MWK Baden-Württemberg within the Juniorprofessuren-Programm, and by the DFG within the SimTech Cluster of Excellence (EXC 310/2).

REFERENCES

1. Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 39–48. DOI:http://dx.doi.org/10.1145/2371574.2371582
2. Ashley Colley and Jonna Häkkilä. 2014. Exploring Finger Specific Touch Screen Interaction for Mobile Phone User Interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design (OzCHI '14)*. ACM, New York, NY, USA, 539–548. DOI: http://dx.doi.org/10.1145/2686612.2686699
3. Hyunjae Gil, DoYoung Lee, Seunggyu Im, and Ian Oakley. 2017. TriTap: Identifying Finger Touches on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3879–3890. DOI: http://dx.doi.org/10.1145/3025453.3025561
4. Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 627–636. DOI: http://dx.doi.org/10.1145/2047196.2047279
5. Seongkook Heo and Geehyuk Lee. 2011. Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 113–122. DOI:http://dx.doi.org/10.1145/2037373.2037393
6. Huy Viet Le, Patrick Bader, Thomas Kosch, and Niels Henze. 2016. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2971485.2971562
7. Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 10. DOI: http://dx.doi.org/10.1145/3173574.3173934
8. Huy Viet Le, Sven Mayer, Patrick Bader, Frank Bastian, and Niels Henze. 2017. Interaction Methods and Use Cases for a Full-Touch Sensing Smartphone. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2730–2737. DOI:http://dx.doi.org/10.1145/3027063.3053196

9. Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2017. A Smartphone Prototype for Touch Interaction on the Whole Device Surface. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 100, 8 pages. DOI: http://dx.doi.org/10.1145/3098279.3122143

10. Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2018. Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI'18)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3173574.3173605

11. Huy Viet Le, Sven Mayer, Katrin Wolf, and Niels Henze. 2016. Finger Placement and Hand Grasp During Smartphone Interaction. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2576–2584. DOI: http://dx.doi.org/10.1145/2851581.2892462

12. Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 220–229. DOI: http://dx.doi.org/10.1145/3132272.3134130

13. Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. DOI: http://dx.doi.org/10.1145/2817721.2817737