A Smartphone Prototype for Touch Interaction on the Whole Device Surface

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

Patrick Bader
University of Stuttgart
Stuttgart, Germany
patrick.bader@vis.uni-stuttgart.de

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

Niels Henze
University of Stuttgart
Stuttgart, Germany
niels.henze@vis.uni-stuttgart.de

Abstract

Previous research proposed a wide range of interaction methods and use cases based on the previously unused back side and edge of a smartphone. Common approaches to implementing Back-of-Device (BoD) interaction include attaching two smartphones back to back and building a prototype completely from scratch. Changes in the device's form factor can influence hand grip and input performance as shown in previous work. Further, the lack of an established operating system and SDK requires more effort to implement novel interaction methods. In this work, we present a smartphone prototype that runs Android and has a form factor nearly identical to an off-the-shelf smartphone. It further provides capacitive images of the hand holding the device for use cases such as grip-pattern recognition. We describe technical details and share source files so that others can re-build our prototype. We evaluated the prototype with 8 participants to demonstrate the data that can be retrieved for an exemplary grip classification.

Author Keywords

Full-touch phone; mobile; capacitive; prototype.

ACM Classification Keywords

H.5.2 [User Interfaces]: Prototyping

MobileHCI '17, September 04–07, 2017, Vienna, Austria.

ACM ISBN 978-1-4503-5075-4/17/09.
https://doi.org/10.1145/3098279.3122143

Introduction

Through a combination of input and output in a single interface, touchscreens became the main interaction mechanism for smartphones. With Back-of-Device (BoD) interaction, researchers formed a field in which the previously unused back of the device is used to perform a wide range of use cases. Amongst others, BoD interaction can be used to perform touch input on small devices without occluding relevant content [2], perform unlock gestures on the back of the device to complicate shoulder surfing [5], or use a BoD touch panel to move the screen content to facilitate one-handed interaction [10]. Similarly, sensors on the device edges enable use cases such as switching applications or navigating on a map [9] and using taps to trigger an application launcher [14]. In previous HCI research, sandwich prototypes and building touch devices from scratch are common approaches to building BoD smartphone prototypes [2, 4, 5]. The disadvantage of these approaches are either a thicker and heavier form factor when attaching devices back to back, or the lack of an established operating system (OS) to integrate novel interaction methods or use cases in well-known applications and user interfaces.

In this work, we present a smartphone prototype that registers capacitive touch input on the whole device surface with a form factor nearly identical to a standard smartphone (i.e. Google Nexus 5). Based on components of a Nexus 5, an established OS is available to e.g. integrate novel interaction techniques into well-known applications. Through Android kernel modifications, the prototype also provides capacitive images that show how the hand touches and holds the device. We describe technical details and provide source files (e.g. PCB schemes, 3D models, and kernel source files) for the reader to re-create our prototype for their future work. We show an evaluation to demonstrate the data that can be retrieved by the prototype.

Use cases that require touch input and capacitive images on the whole device surface include e.g. grip classification to predict actions, authentication, gestures on the BoD and edge, or performing touch input from the back. Further, this prototype can be used for future research to record hand grips in situations where setups such as motion capture systems are not possible (e.g. while walking outside).

Background and Related Work

One approach to implement BoD interaction is to attach an additional smartphone (e.g. [4, 5, 22, 28, 29]) or an external touch pad (e.g. [8, 13, 24]) back to back to the smartphone representing the front screen. While this is a common approach in HCI research [4], the resulting device thickness and weight are noticeably higher compared to commercial devices. These changes to the device's form factor can influence how users hold the device and interact with it. This is detrimental especially for use cases that observe how users hold the device (e.g. context-awareness or prediction) or require users to use front and back of the device simultaneously as shown in previous work [10]. Using two smartphone prototypes with different thickness, Le *et al.* [10] showed that a difference of $3\,mm$ can already affect the grip stability noticeably in a negative way. An unstable grip hampers the finger's freedom of movement as stretching fingers causes the device to drop. Moreover, Sung *et al.* [25] found that the grip span (i.e. the distance between thumb and index finger when grasping an object [23]) has a significant effect on user's touch performance. As the grip span increases with thicker devices, the touch performance decreases which leads to significantly less touch pressure and accuracy during one-handed interaction.

Another common approach is to develop a smartphone prototype from scratch (e.g. [1, 2, 3, 26]). While the resulting form factor depends on the included hardware, the lack of

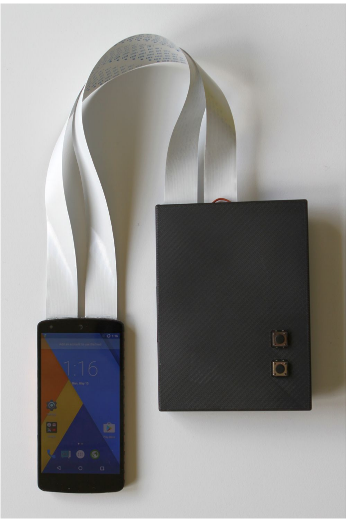


Figure 1: The full prototype with the *Touch Device* and *Hardware Container*.

Figure 2: The *Touch Device* with a front and back touchscreen, and capacitive sensors on the sides.

an established framework (e.g. an OS such as Android or iOS) requires more effort to implement sophisticated use cases such as integrating a novel interaction technique into well known mobile applications or user interfaces. For example, the Android SDK provides a wide range of components, interfaces, and documentation to enable users to easily develop or extend mobile applications. Further approaches to detect touch input on different surfaces include using inaudible sound signals [17, 21, 27], high-frequency AC signals [34], electric field tomography [33], conductive ink printed touch sensors [6], the smartphone's camera [30, 31] and other built-in sensors such as IMUs and microphones [7, 20, 32] to sense touch input. While these techniques do not require any attachments that noticeably increases the device's thickness, their disadvantages make them not suitable for use cases such as grip classification. For example, sound or AC signals are prone to interference while using built-in sensors (e.g. accelerometer or camera) results in limited accuracy or limited area. Further, these approaches do not provide a replacement for capacitive images that are required for use cases mentioned above.

To achieve a familiar form factor, Le *et al.* [10] 3D-printed a back cover for an LG Nexus 5X to attach a resistive touch panel to the back of the device. This increases the device's thickness by only $0.9\,mm$. However, the touch panel being resistive and covering only a quarter of the device makes it infeasible to reconstruct the user's grip. Mohd Noor *et al.* [15, 16] used a custom flexible PCB featuring 24 capacitive sensors distributed around the device's back and sides to maintain a form factor similar to standard smartphones. While they showed that hand grips can be detected adequately, the resolution may still be too low for more precise interaction techniques, such as a 1:1 transfer of a back-touch to the front screen. On the commercial side, the YotaPhone[1] features an e-ink capacitive touchscreen on the rear. However, using front and back touchscreens simultaneously is not possible and edge sensors are not available.

#	Ref-Id	Manufacturer Part No	Data Sheets
2	B2B-F	BM10NB(0.8)-40DS-0.4V(51)	goo.gl/ObJNJk
2	B2B-M	BM10B(0.8)-40DP-0.4V(51)	goo.gl/nWbiIj
3	MPR121	MPR121QR2	goo.gl/iRc4Hq
2	USB-B	Adafruit-1833	goo.gl/SYSW8b
2	BTN	Adafruit-1119	goo.gl/ONkoEr
4	FFC-F	FH12-50S-0.5SH	goo.gl/xtcA2r
2	FFC-C	DIY-08-00073	goo.gl/k3dNvi

Table 1: List of parts comprised in the full-touch smartphone prototype. # indicates the amount required, *Ref-Id* is the identifier we use refer to the respective component.

In summary, previous work presented different types of smartphone prototypes to evaluate novel BoD and edge interaction techniques. While sandwich prototypes and building devices from scratch are common approaches in HCI research, they either change the device's form factor noticeably with negative effects on the hand grip or do not provide the integration into established frameworks.

Smartphone Prototype

To develop a prototype with a form factor similar to an off-the-shelf smartphone (i.e. a Google Nexus 5) while offering the support of an OS (i.e. Android), we used two Nexus 5 smartphones for the front and back side, and a capacitive touch controller to operate additional touch sensors on the sides. We used these components since the touchscreens and side sensors provide capacitive images of the sensed touches so that information such as the finger

---

[1] https://yotaphone.com/us-en/

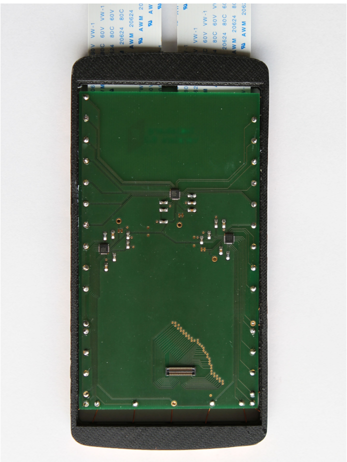


Figure 3: The *Touch Device* module without touchscreens which shows the PCB.

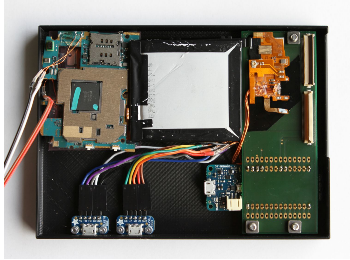


Figure 4: The *Hardware Container* containing the PCB that is connected to the circuit board and batteries of two Nexus 5, and an Arduino to operate the capacitive side sensors.

placement can be retrieved. While we followed the approach of stacking the smartphones, we avoid the increase in thickness by separating the touchscreens from the remaining components of the smartphones. Thus, our prototype comprises two modules (see Figure 1) which we refer to as follows: *Touch Device* represents the smartphone with a front and back touchscreen, and capacitive sensors on the sides as shown in Figure 2; *Hardware Container* is a separated case that comprises the remaining hardware components of both smartphones, including batteries and circuit boards as well as a microcontroller to operate the side sensors (see Figure 4). Each of the two modules includes a self-designed printed circuit board (PCB) which we will refer to as PCB$_{TD}$ and PCB$_{HC}$ for the *Touch Device* and the *Hardware Container* respectively. The PCBs act as extension adapters that are connected to the hardware components within the respective module to establish a connection between the *Touch Device* and *Hardware Container* through flexible flat cables (FFC) as shown in Figure 1. Table 1 lists all required hardware parts, such as connectors or cables. In the following, we describe both modules.

Touch Device

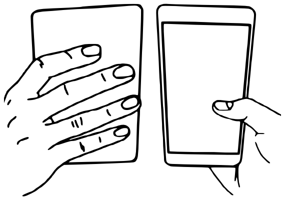
The *Touch Device* (see Figures 2 and 3) consists of a 3D printed frame, two Nexus 5 touchscreens, 37 copper plates as capacitive touch sensors on the edges, and a PCB$_{TD}$. The 3D printed frame holds both touchscreens and encloses PCB$_{TD}$. The capacitive touch sensors are fixated on the left, right and bottom side of the frame. Each touch sensor is a $6 \times y \times 0.5\,mm$ (with $y = 6\,mm$ for left and right, and $y = 12\,mm$ for the bottom side) copper plate which is glued into engravings of the frame with a gap of $1.0\,mm$ in between and sanded for a smoother feeling. The copper plates are connected to the PCB$_{TD}$ which in turn comprises three MPR121 capacitive touch controllers operated by a Genuino MKR 1000 microcontroller in the *Hardware Container*. Similarly, both touchscreens are connected via a board-to-board connector (B2B-F) on the PCB$_{TD}$ and are operated by the remaining components of the Nexus 5 located in the *Hardware Container*. The two FFCs are routed through the top side of the *Touch Device* as there is a low likelihood that it disturbs the user when holding the phone in a usual grip [11]. The dimensions of the *Touch Device* are $137.6 \times 68.7 \times 8.9\,mm$ $(115\,g)$. In comparison, the dimensions of an off-the-shelf Nexus 5 are $137.8 \times 69.1 \times 8.6\,mm$ $(130\,g)$.

Hardware Container

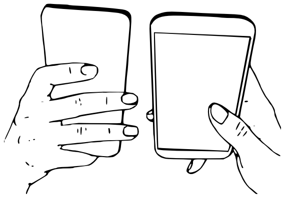
The *Hardware Container* (see Figure 4) is a 3D printed box that contains two Nexus 5 circuit boards and batteries, a Genuino MKR 1000 microcontroller, a micro USB breakout board (USB-B), and two tactile buttons (BTN). When using our prototype in mobile situations such as walking, the *Hardware Container* can be attached to e.g. the user's arm (see Figure 8), belt or placed in a backpack. Thus, we kept the box as small as possible by disassembling the Nexus 5 to remove all frames and unused components (e.g. back camera). The circuit board of the Nexus 5 is connected to a compatible board-to-board connector (B2B-M) on PCB$_{HC}$ which in turn is connected to the touchscreens. To provide access to the Nexus 5's power button and USB port, we replaced the power button and USB port with a replacement button (BTN) and a USB micro breakout board (USB-B) which are placed on the *Hardware Container*. The Genuino MKR 1000 is connected to the PCB$_{HC}$ so that the capacitive side sensors connected to the PCB$_{TD}$ can be operated. The dimensions of the *Hardware Container* are $157.2 \times 115.5 \times 15.2\,mm$ with a weight of $221\,g$.

Retrieving and Processing Touches

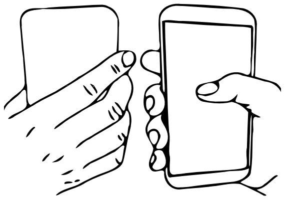
For both Nexus 5, single touch points are accessible through the Android SDK. To retrieve the finger placement for our



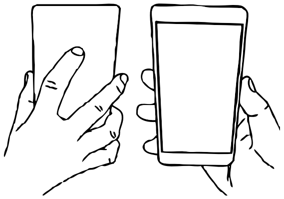
(a) Grip 1



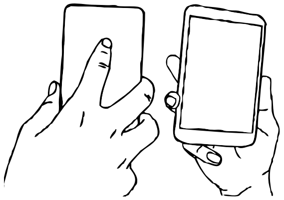
(b) Grip 2



(c) Grip 3

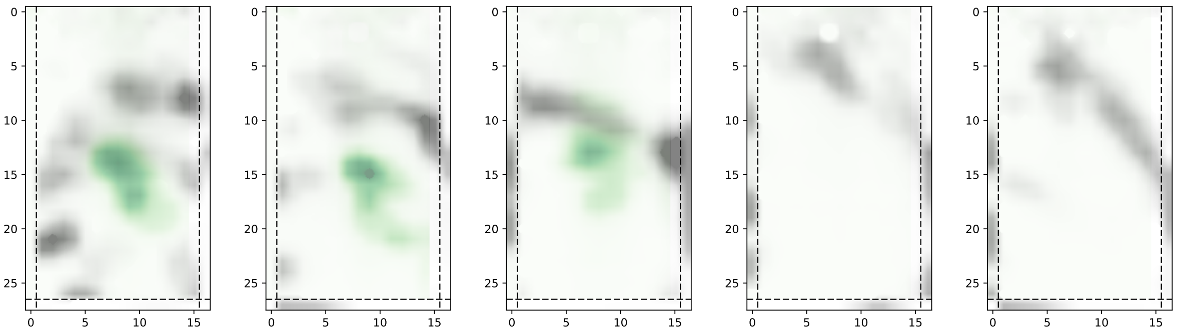


(d) Grip 4



(e) Grip 5

Figure 6: The five different grips from Lee *et al.* [12] as shown to participants.



(a) Grip 1 **(b)** Grip 2 **(c)** Grip 3 **(d)** Grip 4 **(e)** Grip 5

Figure 5: Heat maps depicting the registered touches while holding the smartphone prototype in five different hand grips. The outer rows and columns represent the respective edges of the smartphone. The axes represent the position in mm when facing the front of the smartphone. Green spots represent touches on the front screen, black represents all other sides. These heatmaps are shown from the front perspective.

envisioned use cases, we modified the Android kernel to gain access to the capacitive images reported by the touchscreen (Synaptics ClearPad 3350) controller's debugging interface. These are $27 \times 15\,px$ images in which each pixel corresponds to a $4.1 \times 4.1\,mm$ square on the $4.95''$ touchscreen. The pixel values represent the differences in electrical capacitance (in pF) between the baseline measurement and the current measurement. Capacitive images captured by the prototype are shown in Figure 5. We used I2C calls to access the register for test reporting as described in the RMI4 specification (511-000405-01 Rev.D) and in the driver's source code[2]. Capacitive images are written to the procfs[3] to provide access for Android applications. The MPR121 provides a measurement for each side sensor so that we already have a capacitive image for the sides. The capacitive images are obtained at $20\,fps$ for the touchscreens, and at $130\,fps$ for all side sensors.

To merge the front, back and side capacitive images for analysis (e.g. investigating finger placement), we used a nearest-neighbor approach to match the unix timestamp.

For interactive use cases, the back touchscreen device and the Genuino both transfer their capacitive images to the front touchscreen device which hosts a WiFi hotspot. The transfer latency measured by an average round trip time over 1000 samples is $7.2\,ms$ ($SD = 2.6\,ms$).

Technical Demonstration

To demonstrate the data that can be retrieved from the prototype, we conducted a user study to let participants hold the device using five grips presented in previous work [12].

Participants and Procedure

We recruited 8 participants with an average age of 26.3 ($SD = 2.1$) from the campus of our university. We showed participants five different grips (see Figure 6) in a counterbalanced order using a Latin square. We instructed them to imitate the grip which was shown in a front and back view and to press a button on the front touchscreen to start the recording when they were ready. The capacitive images were recorded for 30 seconds during which participants did not move their fingers. The study took about five minutes.

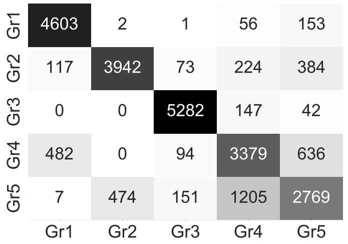
Results

After synchronizing all sides, we have 24,223 frames representing the finger placements. The heatmaps in Figure 5

---

[2] github.com/CyanogenMod/android_kernel_lge_hammerhead/blob/cm-14.1/drivers/input/touchscreen/touch_synaptics_ds5.c

[3] tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html



	Gr1	Gr2	Gr3	Gr4	Gr5
Gr1	4603	2	1	56	153
Gr2	117	3942	73	224	384
Gr3	0	0	5282	147	42
Gr4	482	0	94	3379	636
Gr5	7	474	151	1205	2769

Figure 7: Confusion matrix showing the classification results for the exemplary grip classifier.

Figure 8: The *Hardware Container* attached to a participants arm to demonstrate how the prototype can be used in mobile scenarios.

show the average capacitive measurements over all participants for each grip. Using scikit-learn's[4] MLPClassifier with unchanged hyperparameters, the five grips can be classified with an average accuracy of 82.5% ($min = 62.6\%$, $max = 100.0\%$, k-fold cross-validation over all participants). Figure 7 shows the confusion matrix for the evaluation.

Discussion

We presented a smartphone prototype that senses capacitive touch input on all sides. In a user study, we recorded the finger placement in the form of capacitive images of common grips that were described in prior work [12] to create heat maps (see Figure 6). While it is possible to assign each heat map to a grip by eye, we also used basic machine learning to demonstrate the feasibility of grip classification using the capacitive images. The accuracy of the exemplary user-independent grip classification suggests the feasibility and is similar to the ones reported in previous work [3]. To enable others to re-build our prototype, we share the 3D models, PCB schemes, a list of used parts, and the kernel modifications with the community[5].

In the current state, this prototype has two limitations as a tradeoff for a familiar form factor and the support of an established operating system. Compared to the Nexus 5, the back side is entirely flat instead of slightly curved. This is due to the design decision of minimizing the device size and could be solved by minimally increasing the device's width and thickness to add a curve to the left and right side. The second limitation is the flexible flat cables (FFCs) that are routed through the top side of the smartphone. With a weight of only $4\,g$ and an initial length of $500\,mm$ (readily extendable), we observed that the FFCs neither restrict hand movements (e.g. tilting or moving the device) nor apply any noticeable force on the top edge of the device. Previous work showed that fingers are not touching the device's top edge when holding in portrait orientation [11]. Thus, the FFCs are neglectable when using the device in portrait mode during one and two-handed interaction. To evaluate interaction in landscape mode, the PCB schemes, as well as the 3D frame, can be readily modified to route the FFCs to the right or left side. For studies in mobile scenarios (e.g. while walking or being encumbered [18, 19]), the *Hardware Container* can be attached to the user's upper arm as shown in Figure 8. This prototyping approach is not limited to a Nexus 5 so that larger smartphones or tablets can be used for a larger prototype.

Conclusion and Future Work

In this work, we developed a smartphone prototype that registers touch input on the whole device surface based on capacitive sensing. In contrast to prior work, our prototype has a form factor nearly identical to an off-the-shelf Nexus 5 so that users' hand grip and input performance are not influenced by an increased thickness. To evaluate the prototype's functionality, we conducted a user study to collect touch data for five common grips presented in prior work and visualized the finger placement in heat maps.

In future work, we use the prototype for interaction techniques and use cases that require the finger placement and touch input beyond the front touchscreen. Amongst others, this includes BoD and edge input, grip pattern and change recognition, and an investigation of grips and finger placement in mobile situations where motion capture systems, tracking gloves or cameras are not applicable. While the back touchscreen was only used to capture the finger placement in this work, future work could also look into using the second screen on the back to extend the display space or to present information to the opposite.

---

[4] scikit-learn.org

[5] github.com/interactionlab/full-touch-smartphone

ACKNOWLEDGEMENTS This work is supported by the DFG within the SimTech Cluster of Excellence (EXC 310/2) and the MWK Baden-Württemberg within the Juniorprofessuren-Programm.

REFERENCES

1. Bader, P., Schwind, V., Henze, N., Schneegass, S., Broy, N., and Schmidt, A. Design and Evaluation of a Layered Handheld 3D Display with Touch-sensitive Front and Back. In Proc. of NordiCHI '14.
2. Baudisch, P. and Chu, G. Back-of-device Interaction Allows Creating Very Small Touch Devices. In Proc. of CHI '09.
3. Chang, W., Kim, K. E., Lee, H., Cho, J. K., Soh, B. S., Shim, J. H., Yang, G., Cho, S.-J., and Park, J. Recognition of grip-patterns by using capacitive touch sensors.
4. Corsten, C., Daehlmann, B., Voelker, S., and Borchers, J. BackXPress: Using Back-of-Device Finger Pressure to Augment Touchscreen Input on Smartphones. In Proc. of CHI '17.
5. De Luca, A., von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device Authentication on Smartphones. In Proc. of CHI '13.
6. Gong, N.-W., Steimle, J., Olberding, S., Hodges, S., Gillian, N. E., Kawahara, Y., and Paradiso, J. A. PrintSense: A Versatile Sensing Technique to Support Multimodal Flexible Surface Interaction. In Proc. of CHI '14.
7. Granell, E. and Leiva, L. A. Less Is More: Efficient Back-of-Device Tap Input Detection Using Built-in Smartphone Sensors. In Proc. of ISS '16.
8. Hasan, K., Yang, X.-D., Liang, H.-N., and Irani, P. How to Position the Cursor?: An Exploration of Absolute and Relative Cursor Positioning for Back-of-device Input. In Proc. of MobileHCI '12.
9. Holman, D., Hollatz, A., Banerjee, A., and Vertegaal, R. Unifone: Designing for Auxiliary Finger Input in One-handed Mobile Interactions. In Proc. of TEI '13.
10. Le, H. V., Bader, P., Kosch, T., and Henze, N. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In Proc. of NordiCHI '16.
11. Le, H. V., Mayer, S., Wolf, K., and Henze, N. Finger Placement and Hand Grasp During Smartphone Interaction. In Proc. of CHI EA '16.
12. Lee, S., Kyung, G., Lee, J., Moon, S. K., and Park, K. J. Grasp and index finger reach zone during one-handed smartphone rear interaction: effects of task type, phone width and hand length. *Ergonomics* 59, 11 (2016).
13. Löchtefeld, M., Hirtz, C., and Gehring, S. Evaluation of Hybrid Front- and Back-of-device Interaction on Mobile Devices. In Proc. of MUM '13.
14. McGrath, W. and Li, Y. Detecting Tapping Motion on the Side of Mobile Devices by Probabilistically Combining Hand Postures. In Proc. of UIST '14.
15. Mohd Noor, M. F., Ramsay, A., Hughes, S., Rogers, S., Williamson, J., and Murray-Smith, R. 28 Frames Later: Predicting Screen Touches from Back-of-device Grip Changes. In Proc. of CHI '14.
16. Mohd Noor, M. F., Rogers, S., and Williamson, J. Detecting Swipe Errors on Touchscreens Using Grip Modulation. In Proc. of CHI '16.

17. Nandakumar, R., Iyer, V., Tan, D., and Gollakota, S. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In Proc. of CHI ’16.
18. Ng, A., Brewster, S. A., and Williamson, J. H. Investigating the Effects of Encumbrance on One- and Two- Handed Interactions with Mobile Devices. In Proc. of CHI ’14.
19. Ng, A., Williamson, J., and Brewster, S. The Effects of Encumbrance and Mobility on Touch-Based Gesture Interactions for Mobile Phones. In Proc. of MobileHCI ’15.
20. Roudaut, A., Baglioni, M., and Lecolinet, E. TimeTilt: using sensor-based gestures to travel through multiple applications on a mobile device.
21. Ruan, W., Sheng, Q. Z., Yang, L., Gu, T., Xu, P., and Shangguan, L. AudioGest: Enabling Fine-grained Hand Gesture Detection by Decoding Echo Signal. In Proc. of UbiComp ’16.
22. Shen, E.-l. E., Tsai, S.-s. D., Chu, H.-h., Hsu, Y.-j. J., and Chen, C.-w. E. Double-side Multi-touch Input for Mobile Devices. In Proc. of CHI EA ’09.
23. Shivers, C. L., Mirka, G. A., and Kaber, D. B. Effect of Grip Span on Lateral Pinch Grip Strength. *Human Factors* 44, 4 (2002).
24. Sugimoto, M. and Hiroki, K. HybridTouch: An Intuitive Manipulation Technique for PDAs Using Their Front and Rear Surfaces. In Proc. of MobileHCI ’06.
25. Sung, K., Cho, J., and Freivalds, A. Effects of grip span in one-handed thumb interaction with a smartphone, In Proc. of HFES ’16.
26. Taylor, B. T. and Bove, V. M. The Bar of Soap: A Grasp Recognition System Implemented in a Multi-functional Handheld Device. In Proc. of CHI EA ’08.
27. Wang, W., Liu, A. X., and Sun, K. Device-free Gesture Tracking Using Acoustic Signals. In Proc. of MobiCom ’16.
28. Wolf, K., Müller-Tomfelde, C., Cheng, K., and Wechsung, I. Does Proprioception Guide Back-of-device Pointing As Well As Vision?. In Proc. of CHI EA ’12.
29. Wolf, K., Müller-Tomfelde, C., Cheng, K., and Wechsung, I. PinchPad: Performance of Touch-based Gestures While Grasping Devices. In Proc. of TEI ’12.
30. Wong, P. C., Fu, H., and Zhu, K. Back-Mirror: Back-of-device One-handed Interaction on Smartphones. In Proc. of SA ’16.
31. Xiao, X., Han, T., and Wang, J. LensGesture: Augmenting Mobile Interactions with Back-of-device Finger Gestures. In Proc. of ICMI ’13.
32. Zhang, C., Guo, A., Zhang, D., Southern, C., Arriaga, R., and Abowd, G. BeyondTouch: Extending the Input Language with Built-in Sensors on Commodity Smartphones. In Proc. of IUI ’15.
33. Zhang, Y., Laput, G., and Harrison, C. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In Proc. of CHI ’17.
34. Zhang, Y., Zhou, J., Laput, G., and Harrison, C. SkinTrack: Using the Body As an Electrical Waveguide for Continuous Finger Tracking on the Skin. In Proc. of CHI ’16.