BackTrack: 2D Back-of-device Interaction Through Front Touchscreen

CHANG XIAO, Columbia University
KARL BAYER, Snap Inc.
CHANGXI ZHENG, Columbia University
SHREE K. NAYAR, Snap Inc.

We present *BackTrack*, a trackpad placed on the back of a smartphone to track fine-grained finger motions. Our system has a small form factor, with all the circuits encapsulated in a thin layer attached to a phone case. It can be used with any off-the-shelf smartphone, requiring no power supply or modification of the operating systems. BackTrack simply extends the finger tracking area of the front screen, without interrupting the use of the front screen. It also provides a switch to prevent unintentional touch on the trackpad. All these features are enabled by a battery-free capacitive circuit, part of which is a transparent, thin-film conductor coated on a thin glass and attached to the front screen. To ensure accurate and robust tracking, the capacitive circuits are carefully designed. Our design is based on a circuit model of capacitive touchscreens, justified through both physics-based finite-element simulation and controlled laboratory experiments. We conduct user studies to evaluate the performance of using BackTrack. We also demonstrate its use in a number of smartphone applications.

CCS Concepts: • **Hardware → Printed circuit boards**; • **Human-centered computing → Interaction devices**.

ACM Reference Format:
Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K. Nayar. 2020. BackTrack: 2D Back-of-device Interaction Through Front Touchscreen. 1, 1 (February 2020), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Since the celebrated release of the first iPhone in 2007, the touchscreen has been a first-class citizen on mobile devices. Serving as the major interface both for displaying visual content and for receiving user input, the touchscreen largely defines how we use the device. In many aspects, however, it also rigidifies our use: we are supposed to grip our phone in one hand, with four fingers responsible for holding it and our thumb and possibly even our other hand interacting with the touchscreen. We must constantly move our thumb out of the way of the screen to avoid occluding its visual content, while our other four fingers stay idle most of the time.

To liberate the finger from touching and occluding the screen, the HCI community has become interested in Back-of-Device (BoD) interaction, and various systems have been proposed. For example, by taking advantage of the sensors on the phone, such as the camera [26], microphone [36, 39], and inertial measurement unit [9, 18, 34], the user can deploy fingers that would normally stay idle to tap the backside of the phone and interact with the phone. BoD interactions aim to give the user more flexibility to use the device, by engaging more fingers and alleviating the occlusion of the front

Authors' addresses: Chang Xiao, Columbia University, chang@cs.columbia.edu; Karl Bayer, Snap Inc., kbayer@snap.com; Changxi Zheng, Columbia University, cxz@cs.columbia.edu; Shree K. Nayar, Snap Inc., snayar@snap.com.

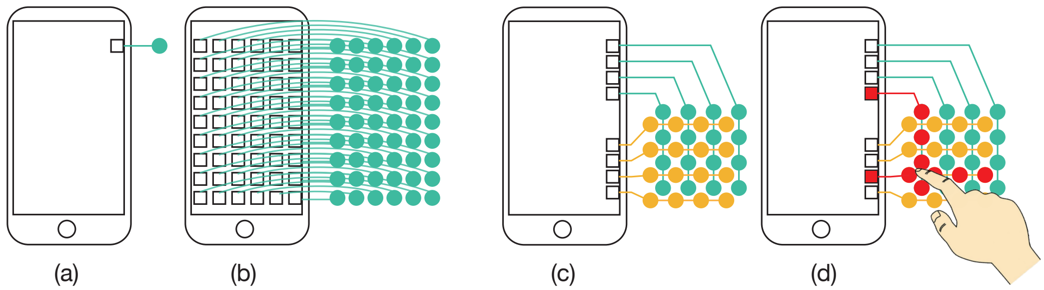

Fig. 2. **BackTrack Overview.** (a) When a touchscreen is connected to a conductive wire, a finger touching at the other end of the wire (i.e., the green dot) will trigger a touch event (under the square) on the screen. (b) A naïve solution attempting to build a 2D trackpad would be to connect every (green) point on the trackpad to a distinct point on the screen through a conductive trace. (c) In our BackTrack design, electrodes on the trackpad are divided into two groups (green and yellow). (d) When a finger touches the BoD trackpad, it will trigger two touch events on the front screen (shown in red). The two touch events indicate, respectively, the column index of the green electrode and the row index of the yellow electrode touched by the finger.

screen. Yet most existing BoD interactions have a limited vocabulary, supporting only finger tapping or 1D swiping (see more discussion in Sec. 2).

In this work, we propose *BackTrack*, a BoD trackpad that tracks fine-grained finger motions in 2D (see Fig. 1). Not only does our system enrich user interaction on the back of the device, it also offers remarkable advantages in daily use. It requires no power supply, no wireless or port connection to the device, and no modification of the operating system—the entire system is encapsulated in a thin layer on the back of a phone case, ready to use with any off-the-shelf smartphones and tablets.

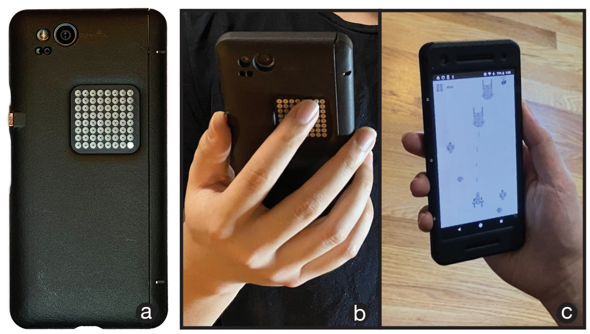

Fig. 1. **BackTrack.** (a) Our BackTrack has a compact and clean design that provides a 2D trackpad on the back of the device. The entire system is encapsulated in a phone case that resembles a conventional case, and requires no power supply. (b) The user can interact with the phone by operating on the back-of-device trackpad. (c) In many apps, this type of interactions avoid occluding the screen.

To design such a trackpad on the back of the device, a natural starting point is the trackpad already on the device, that is, the front touchscreen. Nearly all touchscreens in today's mobile devices are based on *capacitive sensing* techniques. When an electrical conductor, such as a finger, comes into contact with the touchscreen, it distorts the local electric field at that point. This distortion is measured as a change in capacitance, and if strong enough, it is registered as a touch event. Under this working principle, a touch event can be triggered by any conductor of sufficient charge and surface area, not just a finger. For example, one can extend a certain point of the front touchscreen with a long conductive wire; a finger touching at the other end of the wire will trigger a touch event at that point (Fig. 2-a).

BackTrack overview. To leverage this insight for 2D tracking, a naïve approach would be a simple extension of the idea above: connect every point on the trackpad to a corresponding point on the front touchscreen through conductive traces (Fig. 2-b). Thereby, the finger position on the trackpad is one-to-one mapped to individual touch events on the front screen. However, this approach only affords an exceedingly low tracking resolution. If the resolution is $N \times N$, it will consume N^2 points on the front screen and require N^2 traces. It will be very hard to route those N^2 traces in a compact form without any intersection. In addition to being intersection-free, any pair of traces must be well separated. Otherwise, the close proximity of two traces introduces additional capacitance, which will interfere with the measurement of touch-induced capacitance changes.

Instead, we recognize that the front screen is not just a touchscreen but a *multitouch* screen; it can sense capacitance changes at multiple locations simultaneously. Leveraging this feature, we can reduce the design complexity significantly. BackTrack requires only $2N$ traces for a trackpad with resolution $N \times N$, as shown and described in Fig. 2-c. A finger touching on the BoD trackpad triggers two touch events at the same time on the front screen (see Fig. 2-d); those two events encode the x- and y-coordinate of the touching point, respectively.

In our approach, we place $2N$ electrodes near the edge of the front screen so that they can be easily traced to the BoD trackpad (Fig. 2-c). The electrodes are made of indium tin oxide (ITO) coated on a thin glass layer. The ITO glass is fully transparent, covering the front screen just like a screen protector. As a result, the entire front screen continues functioning as it does without the ITO glass. We will present our BackTrack design in detail in Sec. 3, and its use is evaluated by a set of user studies (Sec. 4).

2 RELATED WORK

Back-of-device interaction. Most relevant to our work is the area generally known as Back-of-Device (BoD) interaction. Early work in this area aim to use BoD interaction to avoid finger occlusion of the front screen [12, 17, 28, 32]. More recently, BoD interactions have been used on small-screen devices [2, 35], to augment front-screen operations [3, 5, 25, 37], and for authentication [6, 7, 23].

Many BoD techniques rely on electronic sensors connected to the mobile device [11, 21, 24, 27]. To drive the sensors, a power supply is needed. As a result, these techniques require either a cable connection to the phone to draw power or an external battery. This is a notable limitation. To overcome it, other methods take advantage of the existing sensors already on the phone. Along this direction, the mainstream idea is to leverage physical (in contrast to electronic) channels to communicate between the finger and the device. Various sensing channels, such as light sensed by the camera [26, 33], sound sensed by the microphone [29, 30, 36, 39], and vibration sensed by the inertial measurement unit (IMU) [9, 18, 34, 40] have been explored. However, recognizing intended user interactions from physical raw signals is not as robust as using capacitive sensors, as those signals are often susceptible to environmental noise.

Capacitive sensing [22] has become a technological staple of our daily life, used today on almost every mobile device. We refer the reader to the survey [10] for a thorough review of capacitive sensing in HCI. Here, we only review the works that are closest to ours, and that also extend the capacitive touchscreen with conductive materials. Table 1 summarizes the technical features of our method and closely related works.

Table 1. **Comparison with previous work. IS** indicates interaction space; **FT** indicates whether the method can track continuous finger motions; **OS** indicates if the system is applicable with off-the-shelf smartphone operating systems; **SM** indicates if the system provides a switch mechanism to prevent unintentional touches.

	IS	FT	OS	SM
Clip-on Gadgets [38]	0D	No	Yes	No
ExtensionSticker [16]	1D	Yes	Yes	No
FlexTouch [31]	2D	No	No	No
Ohmic-Sticker [14]	2.5D	Yes	No	No
Kirigami Keyboard [4]	2D	No	Yes	No
CAPath [15]	2D	Yes	Yes	No
BackTrack	2D	Yes	Yes	Yes

Clip-on Gadgets [38] attaches conductive tactile buttons on some locations of a touchscreen. When the user presses a tactile button, it triggers a touch event on the screen. However, the attached buttons will occlude the screen, and the interactions take place only at where the buttons are attached. *Extension-Sticker* [16] further develops this approach by attaching a 1D array of conductive lines to the touchscreen. It allows the user to swipe or tap on a surface outside the screen area, but is limited to 1D operations only. Similarly, *Kirigami Keyboard* [4] maps 1D array of on-screen electrodes to an off-the-phone keyboard, but it cannot track continuous finger motion. *CAPath* [15] uses 3D-printed objects to extend a capacitive touch input, allowing the user to create a customized touch interface. However, the 3D-printed add-on must occlude a large portion of the front screen when it is in use. *Ohmic-Sticker* [13, 14] is based on the principle that the resistance value in a capacitive sensor can also affect the capacitive signal strength. The authors built an input device similar to a trackpoint, and further extended it as a 2.5D input device to sense input forces. *FlexTouch* [31] extends the touchscreen to a large area. It requires access to the touchscreen's raw signal, and leverages a machine-learning model to recognize certain human poses on a yoga mat. Yet it remains unclear how their method can be applied to track finger motions in BoD applications. Both Ohmic-Sticker [13, 14] and FlexTouch [31] require access to the capacitive sensor's raw signals. For off-the-shelf mobile devices, this means kernel modification, which is not always feasible.

Lastly, none of these methods provide a switch mechanism, one that allows the user to enable (or disable) the sensor at will. This may not be an issue for their target applications. For BoD interactions, however, unintentional touches on the back is almost unavoidable. Thus, it is highly desired, if not required, to include a switch to activate the BoD trackpad.

3 BACKTRACK DESIGN

While the basic idea behind BackTrack seems simple (recall Fig. 2), the devil is in the details. Our BackTrack design aims to address several challenges arising from its use in practice:

i) BackTrack should be readily usable on any off-the-shelf mobile devices. Therefore, we can *not* assume the applications using BackTrack have access to the touchscreen's raw capacitance signals. Applications must function using the unmodified operating system's application programming interfaces (APIs). This goal sets us apart from any previous BoD systems that require kernel access (see Sec. 2). We ensure that finger touches on the BoD trackpad to produce strong capacitance changes that will be robustly registered by the unmodified driver as touch events. To this end, we judiciously choose design parameters such as the electrode's size and shape and connection pattern (see Sec. 3.1).

ii) BackTrack encodes a BoD finger touch location in two touch events on the front screen. This encoding scheme, determined by how the trackpad electrodes are connected (Fig. 2-b), has many variations (see Fig. 5 for a few examples). The electrodes' connection pattern will affect the robustness of finger tracking. We therefore need to find one that is robust to commonly used finger motions (see Sec. 3.2).

iii) While the BoD trackpad enriches user interaction, it is also prone to accidental touches—after all, the backside is where fingers are positioned to hold the device. To prevent unintentional touches, we introduce a switch on the side of the device (Fig. 1-a). BackTrack is activated only when the user positions their thumb on the switch. Interestingly, the capacitive nature of the touchscreen allows us to devise a novel switch mechanism that is battery-free and as thin as the copper foil (see Sec. 3.3).

Working principle and design guidelines. To address these challenges, it is essential to first understand the circuit model of BackTrack. We extend the well-understood principle of capacitive sensing [10] to our system, and derive a detailed circuit particularly for BackTrack. The derivation of our model is provided in the supplementary document. Here we summarize several guidelines inferred from the circuit model to shed light on our design.

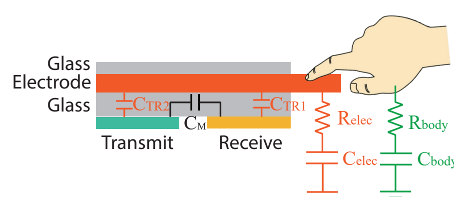

Fig. 3. A circuit diagram of BackTrack. The orange layer here represents an ITO electrode in BackTrack attached on the front screen, the conductive trace, and the connected back electrode. When a finger touches the electrode on the back, it introduces an additional capacitance C_{body}, and thereby triggers touch events at the ITO electrode on the front screen.

Capacitive touchscreen senses the touch of a conductive object, such as a finger, by measuring capacitance change in each pair of

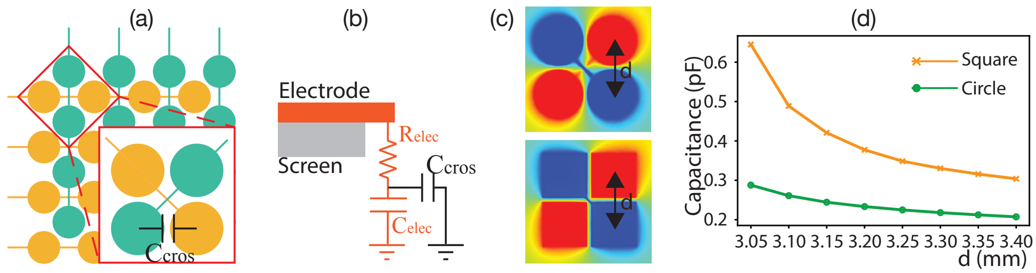

Fig. 4. **Simulation of electrode coupling.** (a) Electrodes that are close to but separate from each other (such as the green and yellow ones in the red box) introduce additional capacitance C_{cros}, which will arise in our circuit model in parallel to C_{elec} shown in (b). Note that here (b) is the circuit model related to one line of connected electrodes (e.g., a row of yellow ones), not to the entire set of electrodes. In (c), We visualize the electric field near the electrodes highlighted in (a) (after applying a 45° rotation). This is simulated using the finite-element method in COMSOL. Here we consider two electrode shapes, circular (top) and square (bottom). (d) We plot the change of C_{cros} with respect to the parameter d (where d is defined in (c)).

electrodes (called *transmit* and *receive* electrodes [10]) placed underneath the touchscreen (i.e., C_{M} in Fig. 3). In our BackTrack design, a number of indium-tin-oxide (ITO) electrodes are placed on top of the touchscreen glass (recall Fig. 2-a). Each ITO electrode, together with the transmit and receive electrodes underneath it, forms two effective capacitors, C_{TR1} and C_{TR2}, respectively (see Fig. 3). Meanwhile, the ITO electrode is connected to the electrodes on the BoD trackpad, which has an additional capacitance C_{elec}.

The existence of C_{TR1}, C_{TR2}, and C_{elec} does not automatically trigger any touch events on the front screen. Although they change the effective capacitance between transmit and receiving electrodes, the change is permanent, and thus the touchscreen driver will not recognize it as a touch event. But when a finger touches some electrodes on the trackpad, the touch introduces another capacitance C_{body} (Fig. 3), which in turn changes the effective capacitance measured by the touchscreen driver. This change is transient, and thus can trigger a touch event.

Our design goal is to ensure that the change introduced by C_{body} is robustly recognized by the driver as a touch event. According to the analysis using our circuit model (in the supplementary document), we seek a BackTrack design that **(i)** reduces C_{elec} while increasing C_{body}, and **(ii)** has C_{TR1} and C_{TR2} as large as possible. In the remainder of this section, we describe our design of electrodes and the switch based on these two guidelines.

3.1 Electrode Design

Electrodes on the BoD trackpad are specified by their shape, size, and spacing d (see Fig. 4). The choices made for these parameters determines the effective capacitance C_{elec} in Fig. 3. According to our circuit model analysis, we wish to reduce C_{elec}. But the reduction of C_{elec} must balance two conflicting factors: On the one hand, electrodes with a small size are preferred, because smaller electrodes introduce less capacitance. On the other hand, as the electrode is reduced in size, it becomes harder for a fingertip to firmly contact the electrode, making the trackpad less reliable. Empirically, we found that a diameter of 3mm is a proper compromise.

Additional capacitance is also introduced due to the proximity of unconnected electrodes. Consider the electrode layout shown in Fig. 4-a. Electrodes in each horizontal (and vertical) line are connected, and across different lines they are separate. As a result, electrodes close to but separate from each other form another effective capacitor C_{cros}, one that arises in the circuit model in parallel to C_{elec} (Fig. 4-b). This effectively increases the electrode's capacitance C_{elec}, and thus we choose electrode shape and spacing with the aim of reducing C_{cros}.

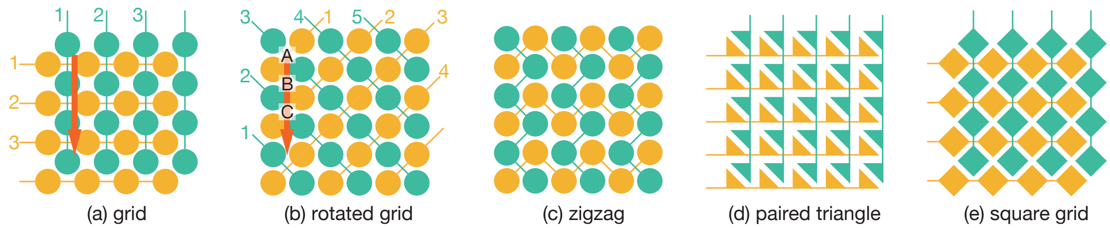

Fig. 5. **Electrode connection patterns.** We show five different patterns for connecting the trackpad electrodes, and refer to them as (a) *grid*, (b) *rotated grid*, (c) *zigzag*, (d) *paired triangle*, and (d) *square grid* patterns. Different patterns encode the coordinate of touch location differently, and lead to different robustness for finger tracking. To understand the robustness difference, consider the grid (a) and rotated grid (b) patterns, and a finger swiping downward (red arrow). With the grid pattern, the triggered row and column indices are (1,1), (2,1), (3,1), . . . When the finger swipes down on the rotated grid pattern from position A (shown in (b)) to C, the triggered index sequence can be either [(1,3), (1,2), (2,2)] or [(1,3), (2,3), (2,2)], and the transition position B will determine which electrode is crossed first. This uncertainty is the source of jittering shown in Fig. 6.

To this end, we use COMSOL, a commercial finite-element simulation tool, to predict C_{cros} under different electrode configurations. As reported in Fig. 4, we consider both square and circular electrodes, each with difference spacings. The simulation shows that circular electrodes lead to a smaller C_{cros} than square electrodes, and as the spacing increases, C_{cros} reduces (Fig. 4-c). Also note that if the spacing d is too large, the electrodes would become too sparse to be contacted reliably. Therefore, we choose to use circular electrodes with a spacing $d = 3.1$mm.

Electrodes on the front screen are mainly specified by their size; their shape and spacing are not crucial. This is because the frontal electrodes are not meant to be directly touched by the finger, and thus they are sparsely placed near the edge of the front screen. The size of a frontal ITO electrode affects the capacitance C_{TR1} (and C_{TR2}) between the ITO electrode and the receive (and transmit) electrode underneath the touchscreen (Fig. 3-c). Enlarging the size will increase C_{TR1} and C_{TR2}. According to our design guidelines, to improve the trackpad robustness, we need to increase C_{TR1} and C_{TR2} and thus enlarge the electrode size.

But the ITO electrode cannot be too large. Otherwise, it will be coupled with many transmit and receive electrodes underneath the screen, and form many effective capacitors. When a finger touches the glass over an exceedingly large ITO electrode, the screen region affected by the finger touch will not depend on the fingertip any more, but rather on the area of the ITO electrode, which is much larger than the fingertip. Then, the finger touch will not be recognized by the driver as a touch event (e.g., the phone will not respond to a contact from our palm). In short, the size of the frontal electrode must not be too large or too small. Empirically, we found that the size of 5mm×12mm (in rectangular shape) with 0.4mm spacing is the most robust. In this setting the front screen still functions as expected.

3.2 Electrode Connection Pattern

BackTrack relies on two groups of electrodes to map a touch location's 2D coordinate into two simultaneous touch events on the front screen (Fig. 2). This mapping is determined by the specific pattern that connects each group of electrodes, and has many options. For example, Figure 5 shows five different patterns. For the purpose of finger tracking, each pattern leads to a different robustness, as illustrated in Fig. 5-a, b and their caption. In what follows, we first evaluate the robustness of various patterns using controlled laboratory experiments, and then propose an algorithm to further improve tracking robustness.

Robustness evaluation. We consider four connection patterns, namely grid, rotated grid, zigzag, and paired triangle patterns (see Fig. 5-a, b, c, d), inspired by the touchscreen's antenna design [1]. For the first three patterns, we have two variants: one with circular electrodes and another with square electrodes. For each pattern variant, we also consider

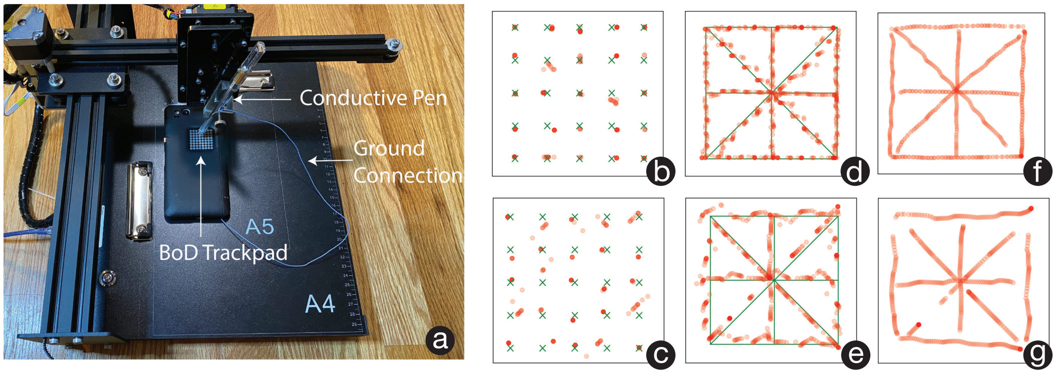

Fig. 6. **Robotic tests.** (a) We use a robotic arm to test the robustness of various electrode shapes and connection patterns on the trackpad. The top row shows the results from the grid pattern in Fig. 5-a, and the bottom row shows the results from the rotated grid pattern in Fig. 5-b. (b, c) visualize the tapping test results: the green cross marks indicate the tapping positions by the robot, and red dots indicate the positions sensed by the BackTrack system. (d, e) show the swiping test results: green lines are swiped by the robot, red dots indicate the sensed swiping trajectories. (f, g) are the swiping trajectories output from the BackTrack system after applying the low-pass filter (1).

two different spacings: $d = 3.1$mm and $d = 3.4$mm (d is defined in Fig. 4-c). In total, we evaluate 14 different methods of electrode connection.

For each method of electrode connection, we evaluate its robustness for two types of user interactions: tapping and swiping. All the evaluations are conducted by a robotic arm holding a conductive pen (whose size is comparable to a fingertip; see Fig. 6-a). In the tapping test, we program the robot to tap 5×5 grid points on the trackpad (Fig. 6-b,c) and measure the mean square error between the sensed touch locations and true locations. In the swiping test, we instruct the robot to swipe along horizontal, vertical, and diagonal lines (Fig. 6-d,e). We then measure the average distance from the sensed touch locations to their intended lines. We repeat each test 5 times and report the average error.

In addition, through an Android kernel hack, we are able to retrieve the raw signal value (also known as the *capacitive image* [21]) returned by the touchscreen driver. This raw value is proportional to the capacitance change of the touchscreen sensor. The detailed results of our experiments are reported in the supplementary document. From these experiments, two patterns stand out in terms of signal strength, namely, the grid and rotated grid patterns both using circular electrodes. The former performs best for sensing finger tapping and swiping horizontally and vertically. The latter is more suitable for finger swiping along diagonal lines. The reason is discussed in the caption of Fig. 5. In practice, tapping and horizontal and vertical swiping are used more often, and therefore we use the grid pattern with circular electrodes in our BackTrack prototype.

Temporal filter. As shown in Fig. 6 and discussed in Fig. 5, swiping along an arbitrary direction may result in a slightly jittered finger path. This is due to the limited trackpad resolution ($N = 7$ in our tests). One can reduce the jittering by increasing the resolution but only to the extent limited by the screen size. Instead, we use a simple low-pass temporal filter to reduce jittering and improve robustness. The low-pass filter is defined as

$$\mathbf{y}_i = \mathbf{y}_{i-1} + \alpha \cdot (\mathbf{x}_i - \mathbf{y}_{i-1}), \tag{1}$$

where the subscript indicates the sensing cycle in time, $\mathbf{y}_i$ is the filtered touch location at timestep i, $\mathbf{x}_i$ is the sensed touch location at timestep i prior to filtering, and α is a parameter that controls the filter strength (in practice, we use $\alpha = 0.1$). This filter is applied in realtime; see Fig. 6 for its effect.

3.3 Switch Design

A BoD trackpad, if not carefully designed, will suffer from unintentional touches. Fingers accidentally touching the trackpad can interrupt the user's interaction with the front screen. To address this issue, we propose a new, compact switch design. We show that a novel use of a piece of copper foil can serve as a battery-free switch. The key idea is based on our circuit model: when C_{body} is sufficiently small, finger touches on the BoD electrodes will not be captured by the touchscreen driver—the trackpad becomes deactivated (see supplementary document for more details).

At this point, our goal is to find a way of switching C_{body} between a low and high value. By definition, C_{body} is the capacitance between our body (i.e., the finger) and the ground. Here a subtle yet important detail is the notion of "ground". It refers to the effective ground of the mobile device, which is not necessarily the physical ground (earth). In fact, when we use a plastic phone case (just like most of us do), the phone's ground is already insulated from our body. Because of this insulation, C_{body} is so small that the BoD trackpad is deactivated. To enable the trackpad, we simply need to connect the user's body with the phone's ground through a conductor, such as a piece of copper foil, so that C_{body} becomes large.

When implementing this switch, we leave a small hole on the side edge of the phone case, near where the thumb is usually positioned. Behind that hole, we place a piece of copper foil connecting to the phone's shell (which is covered by the phone case). When the user wishes to use BackTrack, they just place their thumb over the hole to activate the trackpad. This simple design is battery-free and consumes almost no space. It can also be easily extended by using a momentary mechanical switch to provide haptic feedback or turning it into a toggle switch. To do so, one can apply conductive coatings where the user's hand touches the phone case while holding the phone, and connect the coated part to the phone's ground through a toggle switch. Then, only when the switch is toggled on, Backtrack is enabled.

3.4 Prototype Details

We prototype BackTrack on Google Pixel 2, on which we perform all the experiments including the robotic tests (Sec. 3.2) and user evaluation (Sec. 4). The working principle is also tested on other phone models, including LG Nexus 5, Samsung Galaxy S8, and iPhone 11 Pro Max. To work with Google Pixel 2, we manufacture four parts: a phone case, a frontal glass (1.1mm) coated with ITO electrodes on one side, the side flex PCB, and the back PCB with the trackpad electrodes (see Fig. 7). The frontal glass carries 14 ITO electrodes connected to the flex PCB using conductive epoxies. The other end of the flex PCB is connected to the back PCB. All the components are then encapsulated in a custom-designed phone case, which appears very much like a conventional phone case. The cost of the phone case[1], the frontal glass with ITO coating, the side flex PCB, and the back PCB are $4, $10, $3, and $2, respectively.

Our BackTrack prototype has a trackpad size of 28mm×28mm. It is positioned slightly toward the top-right side on the back of the phone case (Fig. 1); its margins to the top and right sides are 13.5mm and 8.3mm, respectively. This position is at where the index finger is typically placed to host its motion on the trackpad [19]. The trackpad has 7×7 electrodes ($N = 7$), a resolution lower than the front screen. This choice is not a limitation in our design—one can use a higher resolution of electrodes. Rather, this choice is justified by the limited range of finger motions when the finger is positioned on the back holding the device [11, 20]. After all, BackTrack is not meant to replace the front screen, but to enrich it with an additional space for user interaction. Thus, 7×7 electrodes on a small trackpad area suffice.

[1]Here we take into account the material cost for 3D printing the phone case.

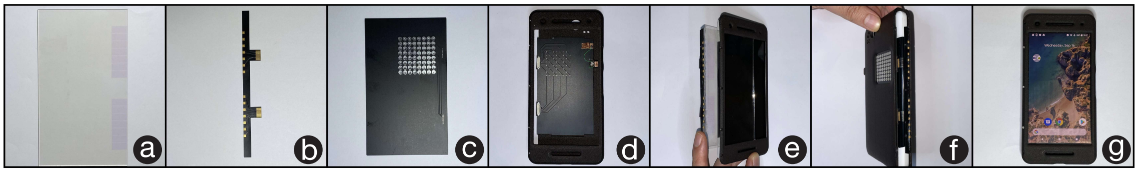

Fig. 7. **BackTrack assembly.** (a) ITO electrodes are coated on the side of a thin glass. The ITO pattern will become fully invisible when attached to the phone's screen. (b) A flex PCB is used for connecting the front ITO and BoD trackpad. (c) BoD trackpad is printed on a PCB. (d) The BoD trackpad is first placed in a custom-designed phone case. We also attach to the BoD PCB a piece of copper foil, which will connect the BoD PCB with the phone's ground after the phone is inserted into the case. This copper foil is also used as a capacitive switch (Sec. 3.3) to enable/disable BackTrack. (e) The ITO-coated glass is connected to the flex PCB by conductive epoxies. We then place the phone together with the glass into the phone case. (f) Lastly, we connect the flex PCB with the BoD PCB. (g) The final fully assembled BackTrack.

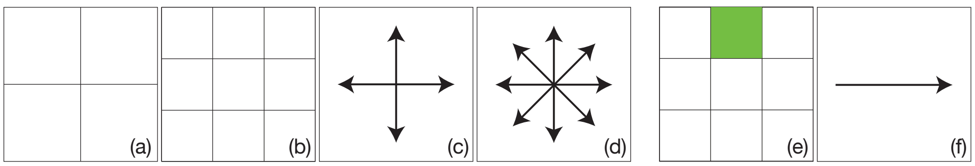

Fig. 8. **Finger motions in user study I.** (a-d) We ask the subjects to perform four tasks. For example, the first task (a) is to tap one of the 2×2 regions on the trackpad, and the fourth task (d) is to swipe along one of the eight directions. (e-f) We display on the front screen what finger motion to perform. The target finger motion is randomly selected each time.

4 USER EVALUATIONS

We conducted user studies that use the proposed trackpads. In particular, we aim to understand what type of finger motions the user can reliably perform on BackTrack. To this end, 12 subjects (6 male, 6 female, aged 22 to 38, 2 left-handed) participated in these studies. None of them had any experience using BackTrack prior to the studies.

Our prototype is designed for the ergonomics of right-handed users—for example, the activation switch is placed on the right edge of the phone to allow the use of the right-hand thumb. We therefore ask the left-handed participants to use their right hand to interact with our BackTrack prototype, although this layout can be easily flipped to adapt to left-handed users.

Methods. Unlike front-screen interaction, finger motions on the back of the device lack visual feedback; the user has little sense of where their finger touches. Further, the user's finger, while positioned on the back holding the device, has a limited range of motion. Thus, only a small area of BoD trackpad is needed and positioned where the index finger is. Our user study seeks to understand to what extent the user can perform finger actions on BackTrack.

We consider two types of commonly used finger motions on a trackpad, tapping and swiping. Our study consists of four tasks, including i) tapping a point on a 2×2 grid, ii) tapping on a 3×3 grid, iii) swiping along horizontal and vertical directions, and iv) swiping along horizontal, vertical, and diagonal directions (see Fig. 8-a,b,c,d).

As none of the subjects had ever used BackTrack, when the user study begins, we inform the subjects that "there is a trackpad on the back of the phone" and ask them to use only their right hands and adjust their grip until they feel comfortable to operate with their index fingers. In each of the four tasks described above, we use the front screen to display what finger motion to perform (see Fig. 8-e,f). The subject then performs the required motion on the BoD trackpad. Each task repeats 50 times; a target finger motion is randomly chosen and displayed each time. Meanwhile, the front screen displays text that tells the subject if their *last* finger motion has been successfully received by the system. This is not a realtime feedback, but feedback as an afterthought: it mimics the type of feedback that the user receives when using BoD interactions in a real-world app.

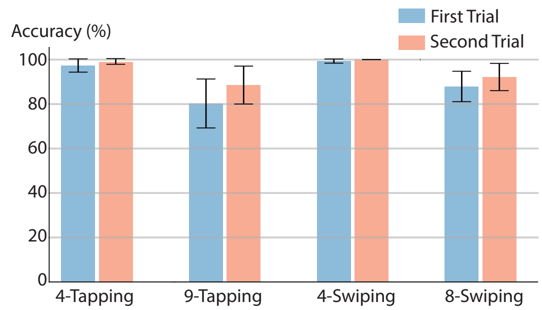

Fig. 9. **User study results.** (a) For each of the four tasks in the first study, we report the finger motion accuracy measured as the ratio of successfully recognized finger motions to the total number of motions. The blue bars indicate human subject performance for the first time they use BackTrack, and the red bars show their performance after three minutes of practice. (b) The System Usability Scale (SUS) results reported by individual subjects.

After finishing all four tasks, we ask the subjects to practice those tasks freely for 3 minutes, and then re-evaluate all the tasks. This is to understand how quickly the user can become familiar with our BackTrack interface.

Results. We report the results in Fig. 9-a, where the accuracy (y-axis) is measured as the ratio of successfully recognized finger motions to the total number of motions. We found that with only three minutes of practice, the accuracies of all four tasks are improved significantly. This suggests that BackTrack is indeed easy to learn. Moreover, the accuracies of 4-way tapping and 4-way swiping are considerably higher than 9-way tapping and 8-way swiping (i.e., 99.16% vs. 88.54% for tapping, and 100% vs.92.17% for swiping in the second trial). This suggests that the mobile app should use 4-way tapping and 4-way swiping as primary ways for the user interacting with BackTrack.

In addition, we are concerned with the ITO electrodes on the front screen affecting the user interaction on the touchscreen, due to the conductive property of ITO that could change the measured capacitance. Therefore, we asked the participants to use the touchscreen as they normally do to see if they experience any abnormality. None of the participants found any noticeable interruption when interacting with the touchscreen, even at regions where the ITO electrodes are placed. Even for tasks requiring high precision such as typing, the participants can still operate smoothly.

5 DISCUSSION

BackTrack enables a wide array of BoD applications, as has been studied in previous works [8, 21]. Inspired by those applications, we built a web browser, a camera app, a game, and a file manager app, to demonstrate BackTrack's robustness and usefulness. We refer the reader to the supplementary video for those demonstrations.

While our goal is to use BackTrack as an add-on device to stock phone models, our method may also be used as a prototyping tool, for studying BoD trackpad design (e.g., with respects to its layout, position, and size), evaluating BoD usability, and for developing BoD applications.

BackTrack shares the same limitations sd other capacitive sensing techniques have. For example, it may not work properly when the user's fingers are wet or wear an insulator, such as a latex glove. It also lacks haptic feedback and does not yet support multitouch. In addition, although our BackTrack design (such as the choices of electrode sizes and shapes) are guided by a circuit model, the guidance remains qualitative rather than quantitative. This is because our goal is to apply BackTrack to stock phone models, and it is hard, if not impossible, to precisely measure the electric attributes (such as the C_{TR1} and C_{TR2}) in the phone's circuits. In the future, we hope to model the BackTrack design in a quantitative way, and develop a systematic design framework for building BackTrack on customized capacitive sensors. It is also interesting to quantitatively analyze how the user conditions, such as the skin moisture and finger size affect the signal-to-noise ratio, under different environments and grounding conditions.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback, as well as Zixiaofan Yang and Anne Fleming for their help on paper revision. This work was supported by Snap Inc. and partly by the National Science Foundation (NSF-1910839).

REFERENCES

[1] Humza Akhtar and Ramakrishna Kakarala. 2014. A comparative analysis of capacitive touch panel grid designs and interpolation methods. In *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 5796–5800.

[2] Patrick Baudisch and Gerry Chu. 2009. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1923–1932.

[3] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. 2014. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1991–2000.

[4] Zekun Chang, Heeju Kim, Kunihiro Kato, Kazuya Saito, Tung D Ta, Weiwei Jiang, Koya Narumi, Yoshinobu Miyamoto, and Yoshihiro Kawahara. 2019. Kirigami Keyboard: Inkjet Printable Paper Interface with Kirigami Structure Presenting Kinesthetic Feedback. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–5.

[5] Christian Corsten, Bjoern Daehlmann, Simon Voelker, and Jan Borchers. 2017. BackXPress: using back-of-device finger pressure to augment touchscreen input on smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4654–4666.

[6] Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. 2014. Now you see me, now you don't: protecting smartphone authentication from shoulder surfers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2937–2946.

[7] Alexander De Luca, Emanuel Von Zezschwitz, Ngo Dieu Huong Nguyen, Max-Emanuel Maurer, Elisa Rubegni, Marcello Paolo Scipioni, and Marc Langheinrich. 2013. Back-of-device authentication on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2389–2398.

[8] Alexander De Luca, Emanuel von Zezschwitz, Ngo Dieu Huong Nguyen, Max-Emanuel Maurer, Elisa Rubegni, Marcello Paolo Scipioni, and Marc Langheinrich. 2013. Back-of-Device Authentication on Smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2389–2398. https://doi.org/10.1145/2470654.2481330

[9] Emilio Granell and Luis A Leiva. 2016. Less is more: Efficient back-of-device tap input detection using built-in smartphone sensors. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. 5–11.

[10] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S Reynolds, and Joshua R Smith. 2017. Finding common ground: A survey of capacitive sensing in human-computer interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3293–3315.

[11] Hiroyuki Hakoda, Yoshitomo Fukatsu, Buntarou Shizuki, and Jiro Tanaka. 2015. Back-of-Device Interaction based on the Range of Motion of the Index Finger. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*. 202–206.

[12] Shigeo Hiraoka, Isshin Miyamoto, and Kiyoshi Tomimatsu. 2003. Behind touch, a text input method for mobile phones by the back and tactile sense interface. *Information Processing Society of Japan, Interaction 2003* (2003), 131–138.

[13] Kaori Ikematsu, Masaaki Fukumoto, and Itiro Siio. 2019. Ohmic-Sticker: Force-to-Motion Type Input Device that Extends Capacitive Touch Surface. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1021–1030.

[14] Kaori Ikematsu and Itiro Siio. 2018. Ohmic-Touch: Extending Touch Interaction by Indirect Touch Through Resistive Objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8.

[15] Kunihiro Kato, Kaori Ikematsu, and Yoshihiro Kawahara. 2020. CAPath: 3D-Printed Interfaces with Conductive Points in Grid Layout to Extend Capacitive Touch Inputs. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (2020), 1–17.

[16] Kunihiro Kato and Homei Miyashita. 2015. Extensionsticker: A proposal for a striped pattern sticker to extend touch interfaces and its assessment. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1851–1854.

[17] Sven Kratz and Michael Rohs. 2009. HoverFlow: expanding the design space of around-device interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–8.

[18] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 321–333.

[19] 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*. 1–10.

[20] 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*. 1–12.

[21] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 779–792.

[22] SK Lee, William Buxton, and KC Smith. 1985. A multi-touch three dimensional touch-sensitive tablet. *Acm Sigchi Bulletin* 16, 4 (1985), 21–25.

[23] Luis A Leiva and Alejandro Català. 2014. BoD taps: an improved back-of-device authentication technique on smartphones. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. 63–66.

[24] Kevin A Li, Patrick Baudisch, and Ken Hinckley. 2008. Blindsight: eyes-free access to mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1389–1398.

[25] Markus Löchtefeld, Christoph Hirtz, and Sven Gehring. 2013. Evaluation of hybrid front-and back-of-device interaction on mobile devices. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. 1–4.

[26] Nobutaka Matsushima, Wataru Yamada, and Hiroyuki Manabe. 2017. Attaching objects to smartphones back side for a modular interface. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*. 51–52.

[27] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 579–588.

[28] Masanori Sugimoto and Keiichi Hiroki. 2006. HybridTouch: an intuitive manipulation technique for PDAs using their front and rear surfaces. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. 137–140.

[29] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 591–605.

[30] Anran Wang and Shyamnath Gollakota. 2019. Millisonic: Pushing the limits of acoustic motion tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[31] Yuntao Wang, Jianyu Zhou, Hanchuan Li, Tengxiang Zhang, Minxuan Gao, Zhuolin Cheng, Chun Yu, Shwetak Patel, and Yuanchun Shi. 2019. Flextouch: Enabling large-scale interaction sensing beyond touchscreens using flexible and conductive materials. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–20.

[32] Daniel Wigdor, Darren Leigh, Clifton Forlines, Samuel Shipman, John Barnwell, Ravin Balakrishnan, and Chia Shen. 2006. Under the table interaction. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 259–268.

[33] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: back-of-device one-handed interaction on smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. 1–5.

[34] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K Nayar. 2019. Vidgets: modular mechanical widgets for mobile devices. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

[35] Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 193–196.

[36] Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott Hudson, and Chris Harrison. 2014. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. 67–76.

[37] Xiang Xiao, Teng Han, and Jingtao Wang. 2013. LensGesture: augmenting mobile interactions with back-of-device finger gestures. In *Proceedings of the 15th ACM on International conference on multimodal interaction*. 287–294.

[38] Neng-Hao Yu, Sung-Sheng Tsai, I-Chun Hsiao, Dian-Je Tsai, Meng-Han Lee, Mike Y Chen, and Yi-Ping Hung. 2011. Clip-on gadgets: expanding multi-touch interaction area with unpowered tactile controls. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 367–372.

[39] Cheng Zhang, Anhong Guo, Dingtian Zhang, Yang Li, Caleb Southern, Rosa I Arriaga, and Gregory D Abowd. 2016. Beyond the touchscreen: an exploration of extending interactions on commodity smartphones. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 6, 2 (2016), 1–23.

[40] Cheng Zhang, Junrui Yang, Caleb Southern, Thad E Starner, and Gregory D Abowd. 2016. WatchOut: extending interactions on a smartwatch with inertial sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 136–143.