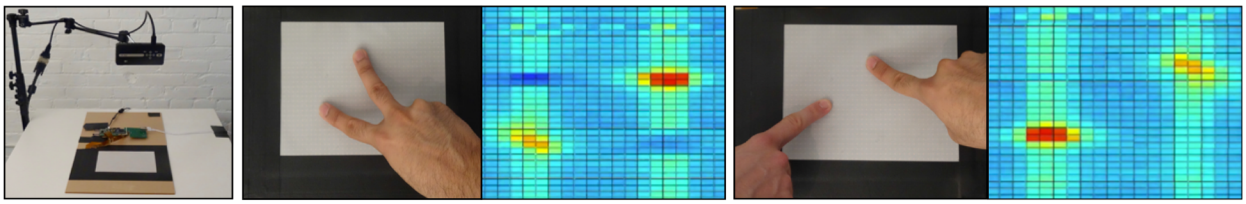
GhostID: Enabling Non-Persistent User Differentiation in Frequency-Division Capacitive Multi-Touch Sensors

Sidharth Sahdev [3,1] **Clifton Forlines** [2] **Ricardo Jota** [1] **Bruno De Araujo** [3,1]
Braon Moseley [2] **Jonathan Deber** [1] **Steven Sanders** [2] **Darren Leigh** [2] **Daniel Wigdor** [3]

[1] Tactual Labs
Toronto, ON, Canada
{first.last}@tactuallabs.com

[2] Tactual Labs
New York, NY, USA
{first.last}@tactuallabs.com

[3] University of Toronto
Toronto, ON, Canada
daniel@dgp.toronto.edu



(a) (b) (c)

Figure 1. (a) *GhostID* prototype with a 10-inch capacitive sensor; (b) two touches from the same hand produce touch points (red and yellow) and ghost signals (dark blue); (c) two touches from two different users produce touch points but no ghosting.

ABSTRACT

Current touch devices are adept at tracking finger touches, but cannot distinguish if multiple touches are caused by different fingers on a single hand, by fingers from both hands of a single user, or by different users. This limitation significantly reduces the possibilities for interaction techniques in touch interfaces. We present *GhostID*, a capacitive sensor that can differentiate the origins of multiple simultaneous touches. Our approach analyzes the signal ghosting, already present as an artifact in a frequency-division touch controller, to differentiate touches from the same hand or different hands of a single user (77% reliability at 60 fps) or from two different users (95% reliability at 60 fps). In addition to *GhostID*, we also develop a framework of user-differentiation capabilities for touch input devices, and illustrate a set of interaction techniques enabled by *GhostID*.

Author Keywords

User Differentiation; Capacitive Touch Sensor; Signal Processing; Mobility

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Interactions

CHI 2017, May 6–11, 2017, Denver, CO, USA.

ACM ISBN 978-1-4503-4655-9/17/05...$15.00.
DOI: http://dx.doi.org/10.1145/3025453.3025719

INTRODUCTION

The commoditization of multi-touch input has enabled device form factors ranging in size from 1″ smart watches to 25′ public displays. While a number of technologies have been used to detect touch input (e.g., resistive, optical, acoustic, etc.), capacitive touch sensing has come to dominate the commercial sphere.

As devices increase in scale, the likelihood and utility of multiple simultaneous users also increases. For example, the research community has repeatedly demonstrated the advantages of single-display groupware in tabletop and large display contexts [3, 10, 26, 27, 38]. Despite the benefits, there remains an important obstacle to commercial deployment in many contexts—the inability, without external hardware, to consistently differentiate between touches belonging to different users [19, 23, 30, 34, 40]. A lack of user differentiation can break well-understood user interface metaphors, such as finger painting [44]: when a single user selects a paint color, that color is then applied to all current users. In addition, further differentiation at the hand level (i.e., do a pair of touches come from two fingers on a single hand, or from one finger on each of a user's hands?) has also been shown to be beneficial [9, 16, 28]. To the authors' knowledge, no commercial devices are able to consistently differentiate multiple simultaneous touches at any level of precision.

To address this limitation, researchers have attempted to design systems that can differentiate touches from multiple users. Some systems augment touch sensing with other sensors, such as cameras [16, 36], fiber optics [12], or proximity sensors [2]. Others augment touch sensors with user-specific signal emitters [7, 18, 19, 34]. While each of these techniques can identify touch origins, the need for additional sensors or external augmentation limits their use cases. An ideal solution for user differentiation would be to enable the capacitive sensor itself to differentiate touches, without the reliance on external hardware.

In this paper, we propose a novel method to differentiate the touches from different hands of a single user, as well as touches from multiple users. The technique is based on the frequency-division multiplexing (FDM) capacitive sensing demonstrated in Leigh et al.'s Fast Multi-Touch (FMT) device [20]. While the primary purpose of FMT was to enable higher frame rates to achieve low-latency sensing, we have found that its unique approach is also well suited to touch differentiation.

A bare capacitive sensor is composed of rows and columns of conductive material. To sense multi-touch input, an FMT system injects orthogonal signals into each row. When a user touches the sensor, their finger will capacitively couple one (or more) rows to one (or more) columns. If a single user touches the sensor in multiple locations, each touch point produces its own coupling of rows and columns. However, because humans are conductive, some of the signals will travel through the hand and into the body of the user. If the user places a second finger on the sensor, these conducted signals will be received alongside the intended signals. The weaker, conducted signals produce faint "ghost" points on the sensor that do not correspond to actual touches.

Leigh et al. view these ghosts as noise, because they do not correspond to actual touches, and therefore filter them out to produce a reliable sensor. However, rather than minimizing and/or simply discarding them, we explicitly use them to differentiate touch origin. If a ghost is present, we know the touch must come from the same user who is already touching the sensor at the position where the ghost signal originated. Moreover, we can use the strength of the ghost signals to detect whether two touches are from the same or different hands of the same user.

The *GhostID* prototype, shown in Figure 1(a), consists of a touch controller implemented on a field-programmable gate array (FPGA) that drives an opaque copper-mesh touch sensor. The controller utilizes FMT principles, except ghosting is not filtered out as noise by the controller.

Our post-processing algorithms can discriminate two touches as coming from the same or different users with 95% precision. Similarly, they can discriminate two touches as coming from one or two hands of a single user with 77% precision. In this paper, we describe our processing pipeline in detail, as well as the results of a study designed to validate the system.

Following the development of our prototype, we designed and implemented a set of interaction techniques that are enabled by *GhostID*. Some of these techniques illustrate the benefits of *GhostID* in a single-user context, while others make use of the ghosting information to support multi-user systems with touch input discrimination.

The primary contributions of this paper are: (1) a framework that defines the space of touch input differentiation; (2) the *GhostID* concept and a prototype that differentiates users and hands on capacitive touch devices without additional hardware; and (3) a set of interaction techniques that are enabled by *GhostID's* capabilities.

A FRAMEWORK FOR TOUCH INPUT DIFFERENTIATION

In conducting a review of the literature on touch differentiation, we found it helpful to classify existing techniques using a three-dimensional space representing precision, persistence, and mobility. In this section, we explain these parameters and classify existing methods within the space. We hope that this framework will aid future researchers in describing their contributions. An overview of the framework is illustrated in Figure 2, and each variable is described in detail below.

Precision

In our framework, we define precision as one of four levels of detail at which a technology is able to differentiate or group touch inputs. The levels of precision are: *tracking*, *user*, *hand,* and *finger*. Each level is cumulative, so a *hand* level technique is also classified as being both *user* level and *tracking* level.

Tracking level precision is the ability to track multi-touch points in real time. The seminal work for tracking multi-touch is by Han [12], who proposed the use of frustrated total internal reflection (FTIR) as the basis of multi-touch sensing. Recent work by Leigh et al. [20] uses frequency-division multiplexing (FDM) to detect multi-touch with extremely low latency.

User level precision is the ability to differentiate between multiple users who are simultaneously using the same touch system. An example is DiamondTouch [7], a tabletop system where users are individually capacitively coupled to the table, providing each with a uniquely identifiable capacitive signature. Other examples include CapAuth [11], Collaid [5], See Me See You [46], Capacitive Touch Communication [40], and HandsDown [36].

Hand level precision is the ability to determine that multiple touches are coming from the same hand. An example is Annett et al.'s Medusa [2], a proximity aware multi-touch tabletop that uses arrays of sensors to track users and hands. Medusa's hand and user tracking technique is fairly precise, but the IR sensors sometimes yield false positives due to reflections from surrounding materials in the room (e.g., pipes, ceiling, etc.).

TouchID [21] uses a wearable glove to track the fingers and hands of users for tabletop interactions. Dhose et al. [8] track users' hands based on skin color and spatial regions around touch points, although their technique suffers from occlusion problems. *GhostID* has *hand* level precision since it can differentiate touches coming from one or two hands from a single user, or the hands of multiple users.

Finger level precision is the ability to differentiate exactly how many fingers are touching a surface, and which hands they belong to. Holz et al.'s Fiberio [16] uses an arrangement of high resolution cameras, fiber optics, and projector-based methods for sensing and identifying multiple users on tabletops. It recognizes an individual user's fingerprints with sophisticated image-processing algorithms. Extended Multitouch [28] uses hand poses to differentiate fingers from different users with a combination of a depth camera and surface touch sensor.

Ewerling et al. [9] use hand and arm images on an optical multi-touch device along with touch information to detect fingers and hands of multiple users. Most biometric fingerprint sensors (like those used to unlock smartphones) have *finger* level precision, but they are not used as general purpose input devices since they are constrained to small patches of specialized capacitive sensing, such as home buttons.

Persistence

We define persistence as the ability of a technology to retain a differentiated touch input. As shown in Figure 2, our framework describes persistence at five levels: *input frame, touch point*, *touch grouping*, *session*, and *lifetime*. As with the precision levels, persistence levels are also cumulative.

Input frame persistence is the minimum persistence required to sense touch. It is the ability to register a touch during a single sensor sampling frame without any continuity or relationship between successive frames. All of today's touch technologies have at least *input frame* persistence.

Touch point persistence is the ability to understand the continuity of a moving touch point between frames as long as the finger remains in contact with the sensor. However, there is no relationship between multiple touch points. Examples include Westerman's [42] methods of tracking multiple fingers and palms on multi-touch surfaces, and McAvinney's Sensor Frame [22] that recognizes gestures in three dimensions. Other examples of *touch point* persistence include Leigh et al. [20] and Han [12].

Touch grouping persistence is the ability to detect that multiple simultaneous touches are caused by the same user. However, this persistence only lasts as long as the user continues to touch the sensor. As one example, Zhang et al. [47] grouped touches as coming from the left or right hands for a single user on a Laser Light Plane (LLP) table. *GhostID* has *touch grouping* persistence.

Session persistence allows *touch grouping* to continue throughout a single usage session even if the user briefly stops touching the sensor. However, if the user stops interacting with the system, leaves, and later returns, the technology cannot identify that the same user has returned. DiamondTouch [7] and Medusa [2] are good examples of *session* persistence. DiamondTouch is persistent as long as the users remain seated in their capacitively coupled chairs. *Session* based differentiation has been extensively studied in the interaction techniques space due in large part to the popularity of the DiamondTouch table as a research tool [26, 37]. Another example is Harrison et al.'s research on Capacitive Fingerprinting [13] that uses Swept Frequency Capacitive Sensing (SFCS), introduced by Santo et al.'s Touché [34]. It detects different users by sensing the electrical properties of the human body and identifying impedance profiles based on variations in bone density, muscle mass, and footwear across individuals. Using a prototype built around the Touché sensor board and integrated with a small touch screen, Harrison et al. also performed a user study involving three demo applications to show the potential of user-aware interaction. More examples of *session* persistent touch input discrimination include See Me See You [46] and Clayphan et al. [5].

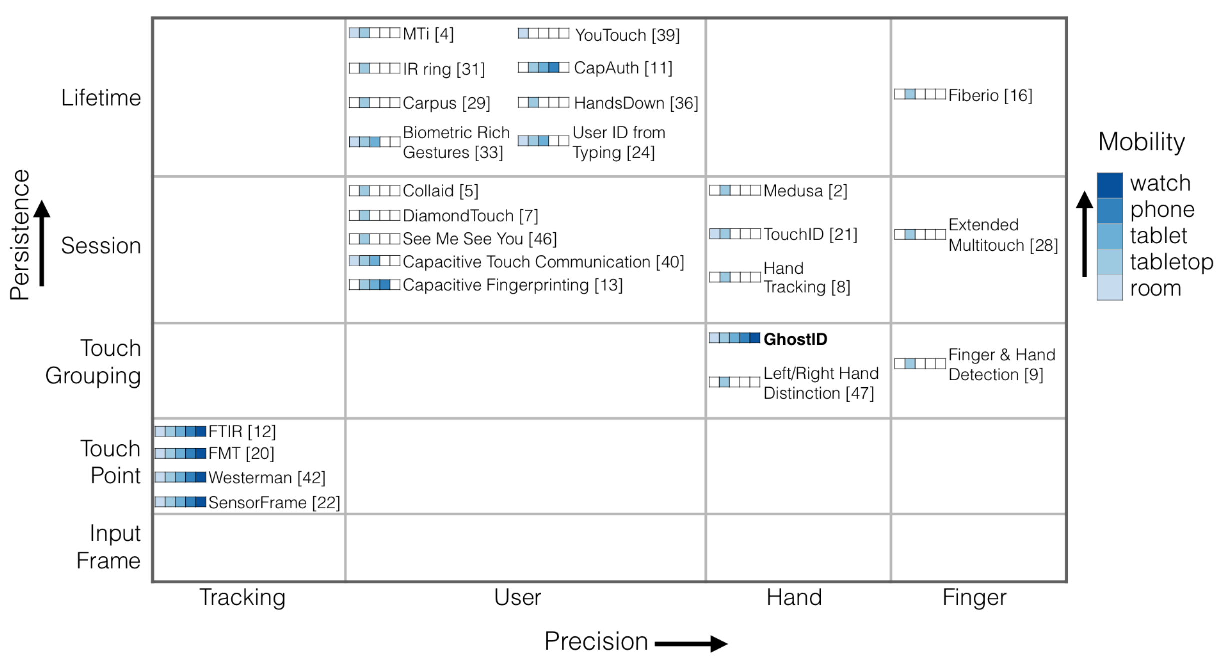


Figure 2. A framework to categorize differentiation techniques. It is parameterized by precision, persistence, and mobility.

Lifetime persistence can indefinitely and uniquely identify a user across all sessions, and can associate all of their touches with them. *Lifetime* persistence is often called *user identification*. In most cases users must perform an initial (implicit or explicit) registration operation when they first interact with the system. Guo et al.'s CapAuth [11] senses changes in the electric fields of the capacitive screen caused by a finger, and can identify users within a smaller group (up to 10 users) at 98.2% accuracy. Although they conclude that their technique is not well suited for high-security applications due to the non-scalability of the technique, the accuracy obtained gives sufficient evidence of the capabilities of capacitance based user-identification.

YouTouch [39] is a recent technique for identifying users on wall-sized displays using a depth camera. There are several other examples of user identification that use external wearable hardware [31], gesture style [33], touch typing style [24], skin texture [29], or hand geometry [4].

Mobility

We define mobility as the scalability and portability of a touch input differentiation technology. These factors fall on a continuous spectrum, but can be roughly divided into five categories in order of decreasing mobility: *smart watches*, *phones*, *tablet*, *tabletops*, and *room*.

The majority of touch input differentiation techniques are low on the mobility scale due to the external support and/or augmentation that they require. For example, Medusa [2] only works indoors since the proximity sensors do not perform well in the presence of sunlight. Likewise, Fiberio [16] is highly precise and persistent, but its integration into smaller form factors is not practical due to the bulky high precision camera that is required. Because it uses a standard capacitive sensor, *GhostID* is highly scalable and can be used across the entire mobility spectrum, ranging from 1″ smart watches to room-sized interactive wall displays.

Other user differentiation technologies

There are also techniques based on external hardware that are capable of differentiating users. These either use wearable hardware or employ user differentiation that does not use finger or palm touches, and therefore they do not fit into our framework [1, 15, 17, 18, 19, 23, 30, 34, 35, 45].

Holz and Knaust's Biometric Touch Sensing [18] seamlessly augments each touch with continuous identification. They rely heavily on a wrist-worn prototype sensor that captures user-unique features and transmits them through the body to a touch screen. Touché [34] is another capacitive touch-sensing technique that uses a small wearable hardware sensor to sense touch. However, the user identification is fragile due to changes in the biometric response to capacitive signals, and returning users are often not identified correctly.

Overall, despite the development of many such approaches, a self-contained (i.e., completely integrated into the capacitive sensor) and accurate multi-user identification system has not been developed.

In summary, *GhostID* is *hand* level precise, has *touch grouping* persistence, and offers a high range of mobility covering smart watches to wall-sized displays. It is capable of differentiating user touches on a large spectrum of capacitive sensors, and is fully integrated into the touch sensor itself.

GHOSTID: USING GHOSTING SIGNALS TO DIFFERENTIATE USERS

GhostID is based on FMT technology [20], and uses frequency-division multiplexing to sense capacitive touch. As noted in the introduction, the touch sensor is comprised of an array of conductive rows and columns. A signal transmitter is attached to each row, and a receiver is attached to each column. The top of the sensor was covered by a 0.8 mm thick sheet of white vellum (used as a top-projection surface), and a glass substrate provided structural support. A photograph of the system is shown in Figure 5a. The dimensions of this stack are 22 cm by 16 cm by 0.1 cm thick, yielding a form factor similar to a 10″ tablet. The sensor contains 40 transmitter rows, and 30 receiver columns. The entire system is powered by a 12V grounded wall adapter. Although we chose to build the prototype using a tablet form factor, the technology can be extended to much larger capacitive touch displays.

FMT transmits a unique, orthogonal sinusoidal signal (3.3 V, 80 – 120 kHz) into each row of the sensor. The rows are capacitively coupled with the receiver columns at every intersection point, and each column receives a baseline level of signal when there are no touches on the sensor. A touch changes the coupling at one or more intersection points, which can be detected by the change in signal intensity. Because the frequencies are orthogonal, multiple frequencies received on a column can be decomposed into their components with a Fourier transform, thereby allowing for a complete snapshot of every row/column intersection during every sampling frame. We refer to this signal snapshot as a *heatmap*. Our prototype is capable of sensing at about 350 Hz. (In comparison, a traditional time-division multiplexing (TDM) touch controller scans the sensor one row at a time, and requires many sampling frames to read the entire sensor; this yields a slower sampling rate. While a ghost-based classification is possible on a TDM controller, the "one-shot" sampling provided by FMT provides a more robust signal.)

A heatmap is computed by the touchscreen controller, and this data is transmitted to a host PC via UDP over Ethernet. Software on the host PC receives these packets, visualizes the heatmap, and performs processing to locate touches and perform the hand and user discrimination.

GhostID gets its name from a signal artifact that occurs when a single user touches the sensor in multiple places. Figure 5c illustrates a heatmap with this artifact. Recall that touching the sensor changes the coupling between a row and a column. With our sensor, we see a decrease in signal intensity since some of the signal is dissipated by the user's body rather than being fed into the receiver. If a single user touches the sensor in multiple places, a signal travels from the first touched row, through the user's body, and is then re-injected into the second touched column. We therefore detect a small increase in intensity of the first row's signal in the second column's receiver, and vice versa. This *crosstalk* does not occur if multiple touches are made by separate people since there is no path for the signal to travel from one person to another.

In Figure 3 (left), we see a single user touching the sensor in two locations. These fingers strongly couple row 1 to column G and row 3 to column D, resulting in two strong touch points at 1-G and 3-D. In addition to these strong pathways, the signal from row 1 passes into the user's right hand, through their body, and into their left hand, which is touching column D. Similarly, some signal from row 3 passes in the opposite direction through to column G. 1-D and 3-G therefore report weaker "ghost" touches that would traditionally be filtered out and discarded as they do not represent actual touch locations.

The strength of these ghost signals depends on the distance the signal must travel between the two touch points. Signals attenuate as they travel through the body, so shorter distances will result in stronger ghosting. If the signal cannot travel between the two points, as is the case in Figure 3 (right) when two different users touch the sensor, there is no ghosting.

Figure 4 shows how this attenuation is used to determine if two touches come from the same hand, from two hands of the same user, or from multiple users. In Path A, the finger directly couples a row to a column with very little attenuation. Path B shows the short distance between two fingers on the same hand. Path C spans from one arm, through the torso to the other arm. Finally, Path D is effectively infinite as there is no electrical connection between start and end points that reside on different users. Through this understanding of attenuation and ghosting, our technique can use the strength of the ghost signals to classify it as coming from one of these paths.

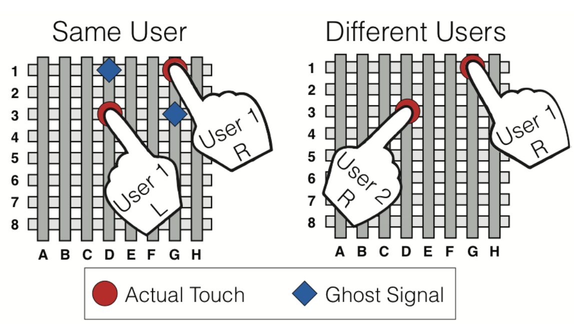


Figure 3. Two touches from the same user produce ghost signals (blue) in addition to real touch signals (red). Two touches from different users do not produce ghosts.

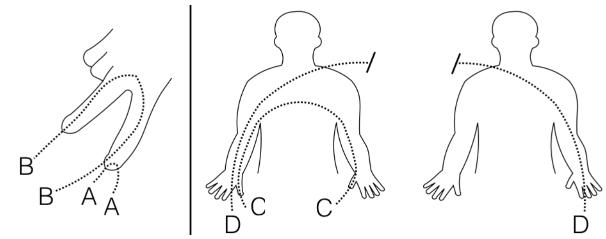


Figure 4. Differing lengths of signal paths (A, B, C, and D) allow user and hand differentiation.

Because our current prototype broadcasts signals on rows and receives signals on columns, it is not able to identify ghosts when two actual touches occur on the same row or column. The strong signals from the actual touch locations will be in the same place as the ghosts, and will therefore mask the weaker ghost crosstalk. This limitation could be overcome with changes to the sensor, as discussed in the Future Work section of the paper.

Signal Processing

This section describes the signal processing pipeline (shown in Figure 6) that enables touch differentiation. The incoming signals are passed through a Python-based software pipeline that consists of: pre-processing, touch detection, ghost detection, feature extraction, per-frame classification, multi-frame analysis, and output of *GhostID* attributes (i.e., touch origin). The *GhostID* attributes are metadata that are associated with each touch point, and are separate from the touch tracking itself; this allows normal touch processing to occur independently of the *GhostID* analysis.

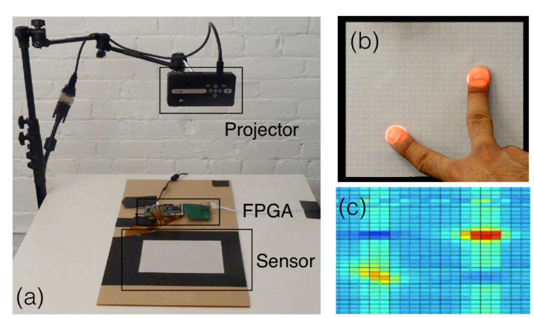


Figure 5. (a) *GhostID* prototype consisting of a 10-inch sensor, FPGA controller, and projector; (b) the sensor surface, with projected feedback; (c) a heatmap visualizing two touch points (red and yellow) from the same hand, and the two resulting ghost signals (dark blue).

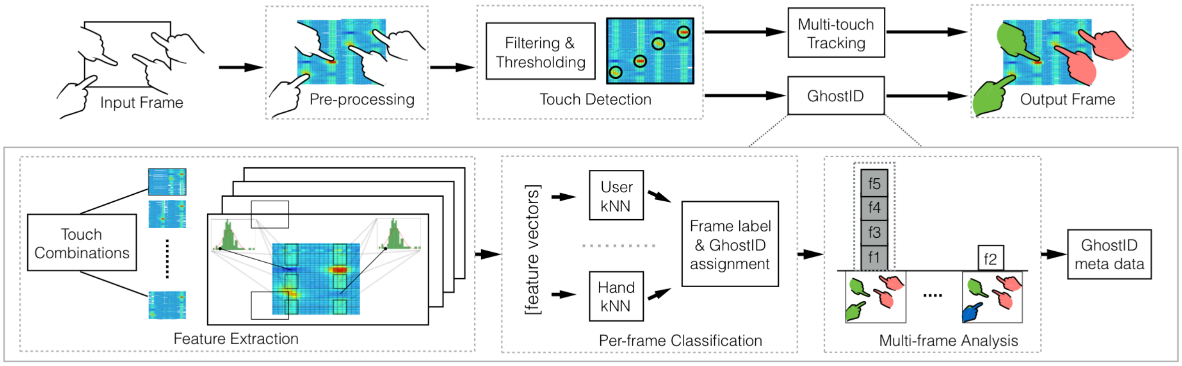


Figure 6. Signal processing pipeline.

Pre-processing. First, each heatmap from the touch controller is checked for validity and completeness. Next, we level the data. Due to several factors, including attenuation of signals across the rows of the sensor, minute variations in adhesive and material thickness and properties, and variations in connections between signal generators and rows, the baseline signal is not uniform across the entire area of the sensor. As such, each pixel (i.e., a row/column intersection in the heatmap) needs to be calibrated to produce a uniform leveling of the sensor data. We record a set of 1000 frames without any touches, and compute an average for each pixel. This average is then subtracted from the incoming raw sensor data to yield a leveled heatmap.

Touch detection. A touch is detected by thresholding the heatmap by signal strength, creating connected components with similar signal intensities, and then finding the location of the local maximum signal strength within each connected component. The threshold level is chosen to remove all ghost signals and leave us with a set of real touches, and is essentially equivalent to the touch detection and ghost removal normally performed by the touch controller. Two outputs are generated from this processing step: touch data is passed out of the pipeline up to the application level, and the heatmap is passed to the remainder of the pipeline to process the ghosts.

Ghost detection and feature extraction. If there are ghost signals, they must by definition appear somewhere along the rows and columns where we have detected real touches (see Figure 3). To search for the ghosts, we generate a list of candidate locations (i.e., a bounding box on the heatmap) by taking each possible pair of real touches and determining the two locations where their ghosts would occur. Next, we examine each ghost candidate by computing a histogram profile of its column, excluding the real touch region and the candidate ghost region. This generates a background noise profile for the column, which follows a normal distribution. We then find the lowest signal strengths within the candidate ghost region, and compute z-scores to compare them to the background noise. All of these features (the signal strength for each touch, and the signal strength and z-score for each ghost candidate) are then fed into classifiers in the next step.

Per-frame classification. We construct feature vectors, and feed them into a pair of binary classifiers (one vs. two users, and one vs. two hands of the same user). We evaluated a variety of classifiers, including k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Decision Trees (iD3), and Logistic Regression. We found that a kNN classifier [6] had the best performance, and made use of a pair of binary classifiers (rather than a multiclass classifier to determine both user and hand classifications) in order to be able to independently tune the features used by each classifier. Our classifiers were trained on a dataset collected with known ground truths; details are provided in the Evaluation section of the paper.

To correctly classify more than two touches, we first classify all possible pairs of two touches using the binary classifiers. Because touch origin is transitive (i.e., if touches A and B are from the same user, and touches B and C are from the same user, we know all three are from the same user), we can assign origins to all touches. This technique scales well with increasing numbers of touch points, and we have tested it for up to four touch points (one touch per user from four users).

Multi-frame analysis. We can further improve the accuracy of the classifiers by performing a classification across multiple frames. Our sensor is capable of producing data at a frame rate of about 350 Hz, and the standard for most displays is 60 Hz. As such, we can group up to five frames together without compromising the standard frame rate for most touch devices. We use a rolling window to group multiple consecutive frames, and a majority voting system tracks the label allotments within each group; this prevents brief classification errors from impacting the final result.

Output of GhostID attributes. We output a set of attributes for each touch point, which allows the application layer to make use of our classifications. Once we output this information, *GhostID* processing can be discontinued until there is a change in the set of touch points (e.g., a new touch point appears), and thereby provide a *touch grouping* level of persistence.

EVALUATION

In order to validate the performance of our prototype and software pipeline, we conducted a study to measure the accuracy of the classifications against known ground truths (i.e., sensor signals captured with known touch origins). We collected a smaller set of initial training data, followed by a larger dataset used for our accuracy evaluation.

Training Data Collection

Our training datasets consists of two parts. The first dataset contained single user data: two touches from two fingers on the same hand vs. two touches from fingers on different hands. The second dataset consisted of data for two users, and contained single touches from pairs of users. We collected both sets of data using 6 participants. All 15 possible combinations of participants were tested in a round-robin design. The data from these experiments was used to build the models used by both classifiers.

Accuracy Study

This study evaluated the accuracy of *GhostID* differentiating touches from one, two, and three simultaneous users, as well as distinguishing one vs. two hands from a single user. Our study consisted of 21 participants (16 males and 5 females) with a mean age of 25 (sd=4) recruited from the local community. Participants were compensated $20. All participants had previous experience with capacitive touch devices.

Apparatus

We used the 10-inch *GhostID* prototype shown in Figure 5a, which is based on an opaque touch sensor with an overhead projector. A desktop workstation with a 3.6 GHz Intel i7 CPU and 16 GB of RAM running Microsoft Windows 10 was used to log the data using a custom Java application.

Trials

The participants were randomly divided into seven groups of three. Each experimental session consisted of a single group of three, and each group performed one session. Participants were seated around the *GhostID* prototype, and each was assigned a color to serve as an identifier. A trial consisted of a standard multi-touch gesture. Each trial lasted for ten seconds, during which time all heatmap data was logged. A color-coded prompt indicated which participant(s) should make what type of gesture. Gestures were illustrated using arrows, and participants moved their fingers along the specified paths, thereby providing consistent gesture sizes.

The selected gestures were traditional multi-touch gestures: a straight swipe gesture with the index finger; a curved swipe gesture with the index finger; a pinch gesture with the index finger and thumb; two finger (index and middle) scroll; and three finger (index, middle, and ring) tap-and-hold. All gestures were performed with the right hand, unless indicated otherwise.

Design

A group session consisted of three parts. The first part involved participants interacting with the sensor individually. The second part involved all three possible pairs of participants interacting with the sensor. The third part involved all three participants interacting with the sensor simultaneously.

Part 1: Single user hand differentiation

Two types of single user gestures were performed: pinch, and two finger scroll. A total of 12 trials were run, which were performed across the entire sensor area.

Pinch. The participants were asked to perform a pinch (back and forth in both directions) using their right index finger and thumb. Participants were then asked to do the same pinch gesture using their right index finger and left thumb. A right index finger and left thumb pinch is not a common gesture; most users would be more likely to use their left index finger for a two-handed pinch. However, we elected to use the left thumb to avoid introducing an additional variable into our comparison, since an index finger would have had a very different touch footprint than the thumb used for the one-handed pinch.

Swipe. The participants were asked to do a two finger swipe back and forth using their right index and middle fingers, and then using their index fingers from their left and right hands.

Part 2: Two-user differentiation

Participants were asked to sit facing each other and perform simultaneous gestures. The positions of the participants were switched half-way through the session to remove any positional bias, and every participant performed gestures across the entire sensor surface. The gestures performed by each participant were: one finger swipe, one finger curved swipe, two finger pinch, and two finger swipe. Because our current prototype cannot perform touch discrimination on touches that are aligned in the same column of the sensor (see Limitations section for a discussion of this issue), we positioned the gestures to avoid this configuration.

Part 3: Three-user differentiation

All three participants were asked to sit around three sides of the sensor and perform simultaneous gestures. Limited physical space on the sensor precluded gestures with a large surface area, so participants performed a one finger swipe. Positions were rotated so that each participant performed the gesture in all four corners of the sensor.

Summary

In total, our dataset contained:

4 gestures × 3 individuals	*(Part 1)*
+ 4 gestures × 2 locations × 3 pairs	*(Part 2)*
+ 1 gestures × 4 locations × 1 triplet	*(Part 3)*
= 40 trials/session	

× 7 sessions (21 participants total)
= 280 trials

× ~2,174 frames/trial
= 608,720 frames

Results

Because each trial has a known ground truth, we can benchmark our classifiers.

When examining a single frame, our hand classifier (one hand vs. two hands) was able to correctly identify the origin of two touch points with a precision of 0.77 (recall = 0.76).

Table 1 shows the performance of our multi-user classifier for two and three touch points. Each column shows the results for one experimental group (each with 3 participants) and each row shows the results for a category of touch points. Averaging across all groups, we were able to differentiate touches from two users 94.9% of the time (sd=10.7%) and three users 98.5% of the time (sd=2.4%); overall precision and recall are shown in the rightmost columns of Table 1. The performance across the groups was observed to be consistent, except for groups 2 and 4. This change in performance is due to perspiration in the hands of one participant in each of those groups (P4 and P10). This moisture was absorbed by the vellum sheet used in the current *GhostID* prototype, and generated spurious signals from the sensor that swamped the relatively weak ghost signals.

If we exclude trials from P4 and P10, we achieve an overall performance of 99.45% (sd=0.6%) for differentiating two touches from two users, and 99.41% (sd=0.5%) performance for differentiating three touches from three different users. Despite the impact of the perspiration on the two participants, we do not consider it a problem moving forward, since the vellum is not a required component of the system. Future iterations will use a glass top layer, and therefore will not be subject to the same effects. Moreover, perspiration is not unique to our techniques. As an example, FTIR [12] is also susceptible to erroneous finger classification due to finger residues if not properly constructed (e.g., a malleable glue overlay of the FTIR surface to evenly distribute the finger pressure and keep finger residues away from the sensing surface).

Our 350 Hz sensor could use a window size (i.e., the number of frames used by the voting system) as large as 5 while maintaining 60 Hz output. In order to select our window size, we performed an analysis using window sizes between 1 and 100. As expected, we see an improvement in performance when using multiple frames, but we do not see significant improvements using more than 4 frames. All of the reported measurers in this paper are therefore based on a 4 frame window. Finally, although we tested our system with three users simultaneously interacting with the sensor, the *GhostID* technology is scalable to many users and is only limited by the device form factor.

	G1	*G2*	*G3*	*G4*	*G5*	*G6*	*G7*	*Precision*	*Recall*
1 user 2 fingers	0.98	0.99	0.92	0.82	0.95	0.88	0.94	0.92	0.94
2 users 2 fingers	0.99	0.71	0.99	0.96	0.99	1.00	1.00	0.95	0.92
2 users 3 fingers	0.86	0.88	0.88	0.64	0.84	0.89	0.97	0.85	0.98
3 users 3 fingers	1.00	1.00	0.99	0.93	1.00	0.99	1.00	0.99	0.86

Table 1. Precision rates for user differentiation for each of our seven session groups. The overall precision and recall across all groups are shown in the rightmost columns.

Techniques to improve differentiation

The accuracy can be further increased with simple heuristics. For example, there are location constraints associated with human anatomy; if touch points are at a sufficient distance from each other, they cannot belong to the same hand. These techniques were not used in the classifier, but future work will leverage this information to improve recognition.

In addition, the accuracy of our single user classifier could be improved with individualized per-user classifier training, since individuals tend to touch with different amounts of pressure and surface area. While this tailoring could not be used in a generic "walk up" situation, it could be useful for personal devices.

ENABLED TECHNIQUES AND SCENARIOS

In this section we discuss some of the interaction techniques we have prototyped using *GhostID*. *GhostID* can extend and enhance some existing techniques that already exist on a variety of sizes of traditional capacitive touchscreens, and can also enable new interaction techniques for mobile devices that are not possible on traditional touch devices, including techniques previously limited to specialized multi-user systems such as the DiamondTouch.

Handling Another Hand

These techniques leverage *GhostID*'s ability to differentiate between single-handed and bimanual interaction by one user.

Single handed mode. The ability to differentiate hands can be used to ignore undesired touches from the non-dominant hand that is holding the device. Accidental activation with the non-dominant hand is very common on most touch devices [43], and can interfere with the user experience. For instance, it can be difficult to hold a larger touchscreen device (e.g., a tablet) by its edges without inadvertently activating UI elements near the sides of the screen. With *GhostID* the user can hold the device without having to adjust their grip to avoid touching the screen. This is particularly useful for edge-to-edge touch screens where fringe contacts cannot be avoided.

Bimanual interactions. In use cases where a single user explicitly intends to interact using both hands, *GhostID* can provide each hand with a different functionality. For instance, the index finger on the dominant hand could be used as a drawing pen while fingers on the non-dominant hand could be used as an eraser or as a mechanism to open a contextual menu [2, 21].

Handling Another User's Hands

GhostID also enables a rich set of multi-user scenarios as depicted in Figure 7.

Ignoring secondary users. People frequently share content on a touchscreen by showing their device to someone else. However, it is quite easy for the second person to unintentionally generate touch input in the process (e.g., by pointing at something) and negatively impact the application (e.g., dismissing a full-screen image, deleting an email message, etc.) [2, 27]. With *GhostID's* ability to distinguish between users, it is possible to ignore the errant touches from the second user. The primary user must still touch the surface with their non-dominant hand to provide continuous *session* level persistence for the duration of the touch, but any additional touch input from other users can now be muted. This allows the secondary user to touch the screen freely without disturbing the contents of the UI.

Handoff. *GhostID* enables a specific interaction technique when handing off a mobile device to another person [14]. If a user passes a mobile device to someone else, as long as both users are in contact with the device for some moment in time (a natural result of handing something to someone else), *GhostID* will be able to tell that a new person is gripping the device. This information can be used to enable collaborative features or provide privacy protection by switching to a guest mode with restricted access. In order to return to unrestricted access, the primary user would need to re-identify themselves, which is not directly supported by *GhostID*.

Collaboration. GhostID provides explicit support for collaboration scenarios. In large displays, where multiple people can interact at the same time, the notion that two hands belong to the same user can help maintain collaboration by specifically supporting actions that only affect one user. As one example, a drawing application can support multiple users painting collaboratively, but any brush or color changes are only applied to the ink stroke that belongs to the user who changed the settings [13].

In addition, *GhostID* enables a wide range of multi-user tabletop techniques at the *touch-user* level of persistence. Many techniques have previously been explored by researchers using the DiamondTouch. Examples include Morris et al. [26] and Shen et al. [37]. We refer the reader to Benko et al. [3], Morris et al. [25], and Ryall et al. [32] for an extensive list of interaction techniques enabled by DiamondTouch and other multi-touch tabletops.

Handling Gestures

Finally, *GhostID* enables extensions and disambiguations of traditional multi-touch gestures.

Disambiguating gestures. We can support a richer set of gestures by leveraging *GhostID's* ability to determine if two points originate from the same hand, different hands of the same user, or different users. From the point of view of a person interacting with a device, the distinction between these three sets of inputs is obvious. However, from the point of view of a traditional touch sensor, those inputs can often appear to be identical. *GhostID* allows the sensor to disambiguate them, and therefore enables richer gestures. For example, two touch points from a single hand can be mapped to the 'resize' command (i.e., a familiar 'pinch' gesture), while two touch points from a user's left and right hands can be mapped to a 'duplicate command' (Figures 8a and 8b). Furthering this example, two touch points from two different users could tear that object in half (Figure 8c) [21, 26]. Traditional touch technologies would have difficulty distinguishing these scenarios.

Gesture granularity. There are also scenarios in which different gestures can be mapped to related actions [21]. For example, consider navigating a map on a large touch surface. With *GhostID*, different granularities can be encoded in the different variations of the same gesture: zooming by pinching with two fingers from the same hand could provide a fine-grain zoom, while pinching using two hands could provide a coarse zoom.

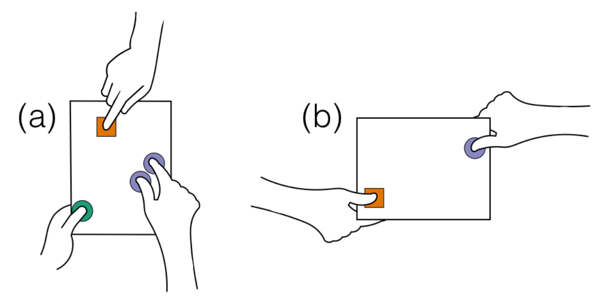


Figure 7. (a) When the primary user interacts using their dominant hand (purple circle), accidental touches from their non-dominant thumb (green circle) or from a second user (orange square) can be ignored. (b) Handing the tablet to someone else triggers a handoff mode with restricted access.

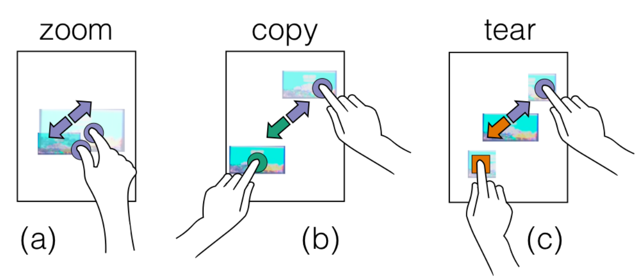


Figure 8. *GhostID* enables disambiguation of otherwise similar touch point paths. (a) A pinch with one hand zooms an image, while (b) two fingers on different hands of one user copies the image, and (c) touches from two different users tears the image.

LIMITATIONS

The current setup is not without limitations. The current implementation only supports a *touch grouping* level of persistence. As we have described in the previous section, this persistence level does enable novel interactions; however, it is not enough for security-based interactions such user identification. From our experience, the ghosting information is not unique to an individual user and, without external hardware that identifies the user, ghosting information alone cannot be used as an identification method.

In order to provide a ghost-free signal for touches originating from two different people, they must be capacitively isolated from each other. While this is the case in most situations, a couple of people can be capacitively coupled by simply holding hands. Once they are in direct contact with each other, signals will travel through both bodies. While in principle we could examine the additional signal attenuation due to the extended signal path and attempt to classify this situation, we have not yet done so. This limitation is most likely to arise in social use cases (e.g., a couple sitting together on a couch), and might actually open an interesting set of collaborative gaming applications [41]. Although we have yet to formally quantify capacitive coupling between individuals, our experience is that indirect contact (e.g., two people leaning on the same wooden table) does not have any impact on *GhostID*. Similarly, grounding a user will create a lower amount of ghosting since some of the signals will be dissipated before they can be re-injected. While we have found that the reduction in signal associated with typical activities (e.g., standing with bare feet on carpet) does not have any impact on our classification, a user who is thoroughly grounded (e.g., touching a grounded metal table frame) will not produce sufficient ghosts.

The current prototype cannot run the *GhostID* classification on touches if more than one of those touches occur in the same row or column because any potential ghosting information is swamped by the signal from the actual touch. We can still register a traditional touch event; we are simply unable to detect if it has a ghost. In practice, this is not a significant limitation, since the lack of ghosting is transient. The ghosting only disappears during individual frames when the fingers are precisely aligned; as soon as they diverge, the ghosts reappear. A brief disappearance of ghosting during a gesture can therefore be mitigated by using a larger rolling window or other heuristics (e.g., positional information). In addition, this issue will be eliminated by forthcoming hardware improvements described below.

Finally, our current prototype sensor suffers reduced accuracy if a user has particularly sweaty hands due to moisture effects on the topmost layer of the sensor. This limitation is unique to the current prototype, and will not be present on future iterations using impermeable top layers.

FUTURE WORK

As future work, we identify two paths of hardware development. First, our sensor needs to be integrated with improved top layer materials, such as glass, or integrated solutions such as ITO. This will strongly reduce the effects of perspiration, and is a necessary step for technology adoption. An ITO-based version of our prototype has been stood up in our lab and, while not ready as of the time of this writing, is showing great promise. Second, our current sensor broadcasts signals on rows and receives signals on columns. This arrangement generally works well, but as discussed above, it cannot detect ghosting information if multiple touches are aligned in the same row or column. A more sophisticated design would eliminate this issue by making several changes. First, each row and column would be connected to a high-speed switch, which would allow the row or column to flip between transmitter or receiver. We then configure the switches so that transmitters and receivers are alternated on both the rows and columns. Finally, we use the switches to flip the transmitter/receiver status of every row and column several times during each frame. This sensor configuration will produce richer ghosting information (helpful information that would allow us to simplify the complexity of the classification algorithms), but also would be able to work around many touch conformations in which touches mask ghosts.

From a software perspective, we intend to explore additional classifiers to further improve our accuracy rates. We also intend to explore additional heuristics to improve both our classification accuracy (e.g., distance between touches), as well as *GhostID's* level of persistence (e.g., time-based heuristics for brief touch removals).

CONCLUSION

We present *GhostID*, a capacitive sensing technique that can distinguish touches from multiple users and even touches from different hands of the same user. Our approach leverages the existing signals from an FDM touch controller, and does not require additional hardware or changes to the sensor itself, making it a potential drop-in replacement for sensing techniques available in the current generation of commercial devices. *GhostID* is presented in the context of a new framework that categorized touch input differentiation. In addition, the value of *GhostID* is illustrated through a description of interaction techniques that are enabled by touch origin differentiation.

Finally, we have just scratched the surface of potential applications for user and hand discrimination. We envision single-user interfaces using hand-discrimination that greatly expand the communication between user and machine, as well as multi-user interfaces that enable simultaneous multi-user interactions that "just work" as expected.

ACKNOWLEDGMENTS

We would like to thank Kate Dowd, and all of our colleagues at Tactual Labs and at the Dynamic Graphics Project lab at the University of Toronto.

REFERENCES

1. Christopher Ackad, Andrew Clayphan, Roberto Martinez Maldonado, and Judy Kay. 2012. Seamless and Continuous User Identification for Interactive Tabletops Using Personal Device Handshaking and Body Tracking. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 1775–1780. http://dx.doi.org/10.1145/2212776.2223708
2. Michelle Annett, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2011. Medusa: A Proximity-aware Multi-touch Tabletop. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 337–346. http://dx.doi.org/10.1145/2047196.2047240
3. Hrvoje Benko, Meredith Ringel Morris, A. J. Brush, and Andy Wilson. 2009. *Insights on Interactive Tabletops: A Survey of Researchers and Developers*. Technical Report. Microsoft Research.
4. Bojan Blažicaa, Daniel Vladušiča, and Dunja Mladenić. 2013. MTi: A Method for User Identification for Multitouch Displays. *International Journal of Human-Computer Studies* 71, 6 (2013), 691–702. http://dx.doi.org/10.1016/j.ijhcs.2013.03.002
5. Andrew Clayphan, Roberto Martinez-Maldonado, Christopher Ackad, and Judy Kay. 2013. An Approach for Designing and Evaluating a Plug-in Vision-based Tabletop Touch Identification System. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13)*. ACM, New York, NY, USA, 373–382. http://dx.doi.org/10.1145/2541016.2541019
6. Padraig Cunningham and Sarah Jane Delany. 2007. *k-Nearest Neighbour Classifiers*. Technical Report UCD-CSI-2007-4. University College Dublin.
7. Paul Dietz and Darren Leigh. 2001. DiamondTouch: A Multi-user Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. ACM, New York, NY, USA, 219–226. http://dx.doi.org/10.1145/502348.502389
8. K. C. Dohse, Thomas Dohse, Jeremiah D. Still, and Derrick J. Parkhurst. 2008. Enhancing Multi-user Interaction with Multi-touch Tabletop Displays Using Hand Tracking. In *Proceedings of the First International Conference on Advances in Computer-Human Interaction (ACHI 2008)*. IEEE Computer Society, Washington, DC, USA, 297–302. http://dx.doi.org/10.1109/ACHI.2008.11
9. Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. 2012. Finger and Hand Detection for Multi-touch Interfaces Based on Maximally Stable Extremal Regions. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*. ACM, New York, NY, USA, 173–182. http://dx.doi.org/10.1145/2396636.2396663
10. Clifton Forlines and Chia Shen. 2005. DTLens: Multi-user Tabletop Spatial Data Exploration. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 119–122. http://dx.doi.org/10.1145/1095034.1095055
11. Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 59–62. http://dx.doi.org/10.1145/2817721.2817722
12. Jefferson Y. Han. 2005. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 115–118. http://dx.doi.org/10.1145/1095034.1095054
13. Chris Harrison, Munehiko Sato, and Ivan Poupyrev. 2012. Capacitive Fingerprinting: Exploring User Differentiation by Sensing Electrical Properties of the Human Body. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 537–544. http://dx.doi.org/10.1145/2380116.2380183
14. Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang 'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. 2014. Sensing Techniques for Tablet+Stylus Interaction. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 605–614. http://dx.doi.org/10.1145/2642918.2647379
15. Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2869–2881. http://dx.doi.org/10.1145/2858036.2858095
16. Christian Holz and Patrick Baudisch. 2013. Fiberio: A Touchscreen That Senses Fingerprints. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 41–50. http://dx.doi.org/10.1145/2501988.2502021

17. Christian Holz, Senaka Buthpitiya, and Marius Knaust. 2015. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3011–3014. http://dx.doi.org/10.1145/2702123.2702518

18. Christian Holz and Marius Knaust. 2015. Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 303–312. http://dx.doi.org/10.1145/2807442.2807458

19. Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 157–166. http://dx.doi.org/10.1145/2807442.2807481

20. Darren Leigh, Clifton Forlines, Ricardo Jota, Steven Sanders, and Daniel Wigdor. 2014. High Rate, Low-latency Multi-touch Sensing with Simultaneous Orthogonal Multiplexing. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 355–364. http://dx.doi.org/10.1145/2642918.2647353

21. Nicolai Marquardt, Johannes Kiemer, David Ledo, Sebastian Boring, and Saul Greenberg. 2011. Designing User-, Hand-, and Handpart-aware Tabletop Interactions with the TouchID Toolkit. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 21–30. http://dx.doi.org/10.1145/2076354.2076358

22. Paul McAvinney. 1986. *The Sensor Frame – A Gesture-Based Device for the Manipulation of Graphic Objects*. Carnegie-Mellon University.

23. Tobias Meyer and Dominik Schmidt. 2010. IdWristbands: IR-based User Identification on Multi-touch Surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 277–278. http://dx.doi.org/10.1145/1936652.1936714

24. Philipp Mock, Jörg Edelmann, Andreas Schilling, and Wolfgang Rosenstiel. 2014. User Identification Using Raw Sensor Data from Typing on Interactive Displays. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. ACM, New York, NY, USA, 67–72. http://dx.doi.org/10.1145/2557500.2557503

25. Meredith Ringel Morris. 2006. *Supporting Effective Interaction with Tabletop Groupware*. Ph.D. Dissertation. Stanford University.

26. Meredith Ringel Morris, Anthony Cassanego, Andreas Paepcke, Terry Winograd, Ann Marie Piper, and Anqi Huang. 2006. Mediating Group Dynamics through Tabletop Interface Design. *IEEE Computer Graphics and Applications* 26, 5 (Sept 2006), 65–73. http://dx.doi.org/10.1109/MCG.2006.114

27. Meredith Ringel Morris, Anqi Huang, Andreas Paepcke, and Terry Winograd. 2006. Cooperative Gestures: Multi-user Gestural Interactions for Co-located Groupware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1201–1210. http://dx.doi.org/10.1145/1124772.1124952

28. Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. 2012. Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 487–496. http://dx.doi.org/10.1145/2380116.2380177

29. Raf Ramakers, Davy Vanacken, Kris Luyten, Karin Coninx, and Johannes Schöning. 2012. Carpus: A Non-intrusive User Identification Technique for Interactive Surfaces. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 35–44. http://dx.doi.org/10.1145/2380116.2380123

30. Stephan Richter, Christian Holz, and Patrick Baudisch. 2012. Bootstrapper: Recognizing Tabletop Users by Their Shoes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1249–1252. http://dx.doi.org/10.1145/2207676.2208577

31. Volker Roth, Philipp Schmidt, and Benjamin Güldenring. 2010. The IR Ring: Authenticating Users' Touches on a Multi-touch Display. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 259–262. http://dx.doi.org/10.1145/1866029.1866071

32. Kathy Ryall, Clifton Forlines, Chia Shen, Meredith Ringel Morris, and Katherine Everitt. 2006. Experiences with and Observations of Direct-Touch Tabletops. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)*. IEEE Computer Society, Washington, DC, USA, 89–96. http://dx.doi.org/10.1109/TABLETOP.2006.12

33. Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, and Nasir Memon. 2012. Biometric-rich Gestures: A Novel Approach to Authentication on Multi-touch Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 977–986. http://dx.doi.org/10.1145/2207676.2208543
34. Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 483–492. http://dx.doi.org/10.1145/2207676.2207743
35. Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. 2010. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 13–16. http://dx.doi.org/10.1145/1866029.1866034
36. Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. 2010. HandsDown: Hand-contour-based User Identification for Interactive Surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM, New York, NY, USA, 432–441. http://dx.doi.org/10.1145/1868914.1868964
37. Chia Shen, Katherine Everitt, and Kathleen Ryall. 2003. UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. In *Proceedings of Ubiquitous Computing: 5th International Conference (UbiComp 2003)*. Springer Berlin Heidelberg, Berlin, Germany, 281–288. http://dx.doi.org/10.1007/978-3-540-39653-6_22
38. Daniel Vogel and Ravin Balakrishnan. 2004. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 137–146. http://dx.doi.org/10.1145/1029632.1029656
39. Ulrich von Zadow, Patrick Reipschläger, Daniel Bösel, Anita Sellent, and Raimund Dachselt. 2016. YouTouch! Low-Cost User Identification at an Interactive Display Wall. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16)*. ACM, New York, NY, USA, 144–151. http://dx.doi.org/10.1145/2909132.2909258
40. Tam Vu, Akash Baid, Simon Gao, Marco Gruteser, Richard Howard, Janne Lindqvist, Predrag Spasojevic, and Jeffrey Walling. 2012. Distinguishing Users with Capacitive Touch Communication. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*. ACM, New York, NY, USA, 197–208. http://dx.doi.org/10.1145/2348543.2348569
41. Cody Watts, Ehud Sharlin, and Peter Woytiuk. 2009. Matchmaker: Interpersonal Touch in Gaming. In *Proceedings of the 4th International Conference on E-Learning and Games (Edutainment 2009)*. Springer Berlin Heidelberg, Berlin, Germany, 13–24. http://dx.doi.org/10.1007/978-3-642-03364-3_2
42. Wayne Westerman. 1999. *Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-touch Surface*. Ph.D. Dissertation. University of Delaware.
43. Daniel Wigdor, Sarah Williams, Michael Cronin, Robert Levy, Katie White, Maxim Mazeev, and Hrvoje Benko. 2009. Ripples: Utilizing Per-contact Visualizations to Improve User Interaction with Touch Displays. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 3–12. http://dx.doi.org/10.1145/1622176.1622180
44. Daniel Wigdor and Dennis Wixon. 2011. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Elsevier, New York, NY, USA.
45. Andrew D. Wilson. 2010. Using a Depth Camera as a Touch Sensor. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 69–72. http://dx.doi.org/10.1145/1936652.1936665
46. Hong Zhang, Xing-Dong Yang, Barrett Ens, Hai-Ning Liang, Pierre Boulanger, and Pourang Irani. 2012. See Me, See You: A Lightweight Method for Discriminating User Touches on Tabletop Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2327–2336. http://dx.doi.org/10.1145/2207676.2208392
47. Zhensong Zhang, Fengjun Zhang, Hui Chen, Jiasheng Liu, Hongan Wang, and Guozhong Dai. 2014. Left and Right Hand Distinction for Multi-touch Tabletop Interactions. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. ACM, New York, NY, USA, 47–56. http://dx.doi.org/10.1145/2557500.2557525