PhoneTouch: A Technique for Direct Phone Interaction on Surfaces

Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, Hans Gellersen
Computing Department, Lancaster University, Lancaster, UK
{schmidtd, chehimi, rukzio, hwg}@comp.lancs.ac.uk

ABSTRACT
PhoneTouch is a novel technique for integration of mobile phones and interactive surfaces. The technique enables use of phones to select targets on the surface by direct touch, facilitating for instance pick&drop-style transfer of objects between phone and surface. The technique is based on separate detection of phone touch events by the surface, which determines location of the touch, and by the phone, which contributes device identity. The device-level observations are merged based on correlation in time. We describe a proof-of-concept implementation of the technique, using vision for touch detection on the surface (including discrimination of finger versus phone touch) and acceleration features for detection by the phone.

Keywords: Interactive tabletops, surface computing, mobile phones, personal devices, interaction techniques.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. – Input devices and strategies

General terms: Design, Human Factors, Algorithms

INTRODUCTION
When people interact with shared devices such as interactive surfaces, the question arises as to how their personal devices can be brought into play. The purpose of using personal devices in a collaborative setting might be to act as a proxy for their user (such that input events can be associated with different users), to affect control on the shared device, or to transfer data between the personal and the shared context.

In particular, recent work has investigated use of mobile phones in conjunction with interactive surfaces. Techniques have been presented for pairing of phones by placing them on a surface [12], and for localization of phones such that their position and orientation can serve as input [1]. In both cases, interaction is affected by placing a phone on the surface, much like a token. In contrast, we propose a novel technique, *PhoneTouch*, for direct phone interaction on surfaces, where the phone is used like a stylus. This affords fast and fluid interactions on the surface, such as pointing and selection of targets, and copying of objects from the surface to the phone and vice versa. Figure 1 illustrates the user experience: phones can be used alongside conventional touch input, and are uniquely identified when they touch the surface.

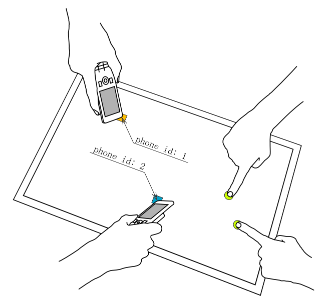


Figure 1: PhoneTouch enables direct phone interaction on surfaces alongside conventional multi-touch.

The PhoneTouch technique is based on distributed sensing and recognition. The interactive surface detects touch events and localizes these in terms of surface coordinates. The mobile phone detects touch events concurrently and associates these with their identity, and possibly with other device states (e.g., a key pressed). The devices communicate detected events in real-time to a central server, where correlation in time is used to identify shared events. Multiple phones can be used in parallel for interaction, as events are resolved to different devices identities, however a conflict can occur when two or more phones touch the surface within the same recognition timeframe. The event correlation abstracts from how surface and phones detect events, which means that sensing methods are not prescribed and can vary among the devices.

PhoneTouch integrates a range of previously demonstrated concepts, to provide a novel way of using mobile phones for interaction with interactive surfaces. The stylus-like use of phones for touch interaction with displays was explored by Hardy et al. but with a coarse grained touch input method (grid of NFC tags) [3]. A related interaction style is Rekimoto's "Pick & Drop" [8]. Distributed sensing of object touch on surfaces, and collaborative inference of "what was placed where" was shown by Strohbach et al. [11]. In other work, similarity of sensor observations has been used for coupling or pairing of devices [5, 4, 6].

The contributions of this paper are as follows. Firstly, we introduce the design and architecture of PhoneTouch. Secondly, a proof-of-concept implementation is described in which we use vision for touch detection on the surface and acceleration features to detect touch events in mobile phones.

Thirdly, results of an experiment are presented in which we verify classification of phone touch versus finger touch on surfaces, and analyze temporal distribution of touch events in a multi-user scenario.

PHONETOUCH DESIGN

User Experience

Figure 2 illustrates the use of PhoneTouch in a scenario. Andy, Bart and Chris meet around an interactive tabletop. One of the friends, Andy, wishes to share a collection of photos he has taken on a recent trip. He takes out his phone, starts the picture sharing application, selects the photos, and then touches the tabletop. The selected photos immediately appear on the table, spread out around the point of contact (Figure 2(a)). He pockets his phone and the three friends start browsing the photos, using their fingers on the multi-touch table (Figure 2(b)). The friends enlarge several of the photos for a closer look at them and arrange them by interest. Bart and Chris take out their phones, also start the picture application, and pick up photos they would like to take home by touching them with their phones (Figure 2(c)).

(a) (b) (c)

Figure 2: A scenario of PhoneTouch interaction: (a) Andy transfers a collection of photos onto the surface. (b) With his friends he is browsing the collection. (c) Chris copies a photo to his phone by touching it.

The user experience of PhoneTouch is defined by the following characteristics:

- The phone is used like a stylus for pointing and object selection on the surface. From a user's perspective, any interaction that can be performed with a single finger touch can in principle also be performed with a phone touch.
- The input space for finger input and phone input are identical, such that phone touches are resolved with the same level of granularity as positions on the surface.
- While the same input is possible with finger or phone, finger and phone events are distinguished, and phone events are associated with the identity of the phone and potentially other phone states, such as holding of keys on the phone as modifier of the touch event.
- The technique seamlessly extends conventional multi-touch on interactive surfaces. There is no restriction or compromise of established styles of tabletop interaction in order to facilitate PhoneTouch.
- PhoneTouch is also seamless in extending mobile phones. Users can move fluidly between use of their phone for touch interaction on the surface, and interaction with the phone itself as personal device.

PhoneTouch is a generic technique that facilitates a range of applications:

- *Data transfer* between mobile devices and interactive surface is supported in a fluid, lightweight, and intuitive manner by direct selection.

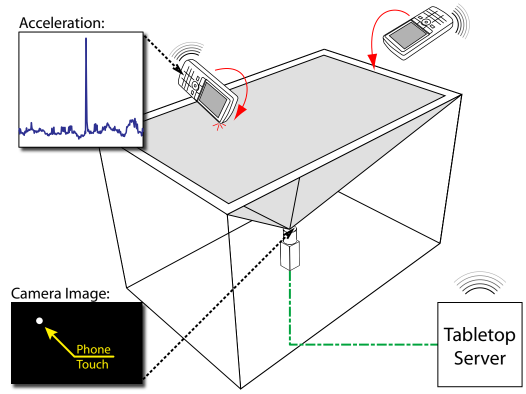


Figure 3: Surface and phones detect touch events independently. The device-level observations are communicated over a wireless network, correlated in time, and combined to associate a touch with both surface position and phone identity.

- User-identified input enables *interface personalization* on the surface (e.g., personalized menus) to facilitate colocated collaboration.
- *Authentication* on shared surfaces is enabled with the phone serving as security token, or by using it as private display to enter a shared secret unobserved.

System Design

The system design for PhoneTouch is illustrated in Figure 3. The principal idea is that all involved devices, the surface and the phones, independently detect touch events. Detected device-level events are time-stamped and communicated in real-time to a server, over a wireless link. The individual surface and phone events are matched based on their time-stamps, in order to determine PhoneTouch events. The PhoneTouch events combine complementary information contributed by surface and mobile device: location of the touch and identity and state of the phone. As the matching is based exclusively on synchronous timing, there is no requirement for use of specific sensors. This principle of using co-occurrence of events in abstraction of sensors has precedents in a variety of works such as *Cooperative Artefacts* [11] and *SyncTap* [9].

As the technique is based on event correlation in time, the system clocks of the surface and the phones need to be pairwise synchronized. When all participating devices already share a network, a network time protocol can be used for synchronization. For initial pairing of phones with a surface, a possible synchronization method, similar to *SyncTap* [9], would be to prompt the user for three successive phone touches on the surface. This would generate two relative time intervals that the devices would share to determine the offset of their clocks.

As PhoneTouch interactions are centered around the interactive surface, a natural communication topology would be to have the surface computer act as central server on which device observations are combined. For general application of the technique, the phones would not need to know the location of the touch, however depending on application needs, device-level events could also be shared peer-to-peer, such that phones obtain the surface location of their own touches and the touches of other phones.

PROOF-OF-CONCEPT IMPLEMENTATION

Phone

Ronkainen et al. [10] showed that accelerometers afford the reliable detection of tapping events on mobile devices. In our initial implementation, we chose to use an externally mounted sensor unit to achieve higher sampling rates than currently possible with built-in accelerometers. In doing so, we attached WiTilt V3 wireless sensors to three Nokia 5800 mobile phones.The integrated 3-axis accelerometer samples at 130Hz and communicates via Bluetooth. On the phone, we run a threshold based detection algorithm which identifies narrow, sharp peaks characteristic for touches.

Table

We use a custom built interactive tabletop with an active surface area of 91cm×57cm and a rear-projected screen with a resolution of 1280px×800px. Touch detection is based on computer vision in conjunction with frustrated total internal reflection (FTIR) [2]. The employed camera has a resolution of 640px×480px and captures images at 120Hz. Any object in contact with the surface is clearly visible after applying highpass, dilate, and thresholding filters. We extract contact areas by identifying connected components.

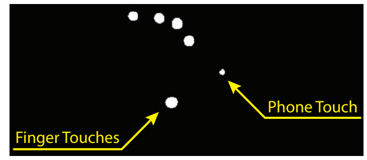
Communication

Phones and interactive tabletop exchange synchronization messages and time-stamped events via Bluetooth, thereby operating in different piconets than the external WiTilt sensors. Pairwise synchronization between phones and tabletop is achieved by a time-stamped message exchange, described in the network time protocol [7]. Round trip times below 10ms result in a maximal clock offsets of 5ms. The tabletop advertises its service over Bluetooth which can be found by mobile devices to form an ad-hoc network. Initial pairing naturally cannot be based on absolute event timing as no clock synchronization has taken place yet. Instead, users are asked to tap three times with their phones, generating relative time intervals which are separately sensed by phones and tabletop for comparison.

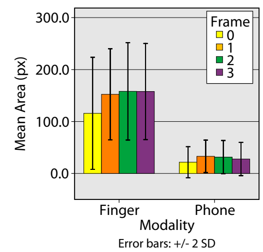
Finger and Phone Discrimination

As depicted in Figure 4(a), a phone touch (right) results in a substantially smaller blob than finger touches (left). Testing different phone models and varying the angle while touching had very little impact on the observed contact area; even touches with the entire edge were readily distinguishable. A pilot study confirmed this observation and suggests that area size is a reliable indicator for distinguishing phones and fingers. In particular, we asked 12 participants (3 female) recruited from our department to successively touch targets appearing on the surface at pseudo-random locations. Participants completed two rounds touching 64 targets first with a phone then with their fingers; the presentation order of interaction type was counter balanced. We observed users holding phones in a variety of ways during trials.

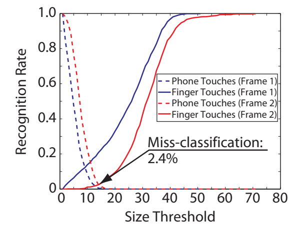
Contact areas were analyzed over the first four frames after touch detection. We observed a high variance in the first frame which can be explained by an approaching phase: initially, the touch area is small and grows until full contact is made. This effect is more pronounced for soft objects like fingers. As Figure 4(b) indicates, there is no substantial improvement in terms of reduced variance after the second frame where mean areas of 152.35 (*SD*=43.83) and 33.04 (*SD*=15.62) were recorded for fingers and phones respectively. As depicted in Figure 4(c) the best trade-off for setting a discriminating threshold in the second frame results in a miss-classification of 2.4%. If reliably detecting all phone touches is the aim, 9.5% of fingers will be miss-classified. The results were observed for adults using standard phones. Phone-finger discrimination might be affected if phones are used with soft protective skins, or if users with smaller fingers (e.g., children) use the system.



(a) Finger and phone touches yield different contact areas (table camera view).



(b) Variances in contact area over the first four frames.



(c) Effect of size threshold on recognition rates of fingers vs. phones.

Figure 4: Finger and phone touches on the surface can reliably be distinguished by contact size.

Event Correlation

Conflicts arise when phone touches collide, i.e. fall into the same recognition timeframe. Ideally, a touch is instantly detected and assigned the exact time of occurrence. However, such a scenario is unrealistic due to finite sampling rates, detection latencies, and clock synchronization offsets. While sampling rate and detection algorithm cause delays, negative clock offsets possibly lead to events appearing too early.

The described prototype has a minimum recognition timeframe of 25ms (max. sample length of 8.33ms+clock offset of 5ms×2, rounded up to the next full sample). We designed detection algorithms for low latency but cannot quantify their delay characteristics at the current stage. It is noteworthy that conflicts are detected. Dependent on the application, suitable measures can be taken to resolve them (e.g., users can be asked to repeat their action).

EVALUATION

The following experiment serves two purposes. First, we verify finger versus phone classification performance on the surface. Second, we analyze the temporal distribution of phone touches in a multi-user task. Participants are asked to select either a single picture out of a set of 9 or multiple pictures

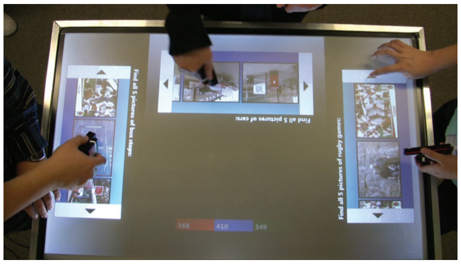


Figure 5: The scroll buttons are activated by finger touches while phone touches select a picture.

out of a set of 14. Figure 5 shows the task layout. Each user is presented with a horizontal scroll list, showing between two and three pictures at a time. Above, we print the current search task. Participants can scroll by touching the arrow buttons on either side with their fingers. To generate a high number of events, we opted for repeated touching to advance the list rather than holding down a button. A picture can only be selected by phone touch.

Twelve participants (7 female), split into groups of three, were recruited from the local campus (mean age of 23.2, $SD = 5.03$) and compensated with £5 for their time. Before beginning the study task, they were given the opportunity to test the system until they felt comfortable using it. While working simultaneously but independently within their surface area, participants completed 20 search tasks each. Contacts with the surface were detected using the computer vision approach of our proof-of-concept implementation.

Results

We observed a mean task completion time of 6:27 minutes. The discrimination threshold was optimized not to miss any phone events, resulting in a correct phone classification rate of 99.99% while miss-classifying 5.66% of fingers as phones, thus exceeding the expected performance predicted in the pilot study.

Further, we measured time differences between successive phone touches. Figure 6 shows the temporal distribution of phone touches, accumulated in intervals of 25ms. With an assumed recognition timeframe of 25ms for our prototype, 97.7% of phone events could be detected without collision, assuming no false positives. Taking into account the fraction of finger touches miss-classified as phones (competing with true phone touches) 96.3% of phone events are still collision-free; potential false positives originating from acceleration-based recognition are not considered. The presented results provide insights into the simultaneity of phone touches and requirements for sensing hardware. However, this evaluation can only serve as an indication since touch frequencies are dependent on application and number of concurrent users.

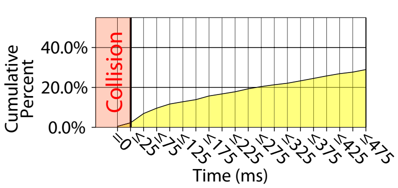


Figure 6: Temporal distribution of phone touches (time differences accumulated in intervals of 25ms).

CONCLUSION

We presented PhoneTouch, a novel technique for integrating mobile phones and interactive surfaces. Based on distributed sensing, this technique enables use of phones to select targets on the surface by direct touch. Further, we described a proof-of-concept implementation. An initial evaluation with focus on finger versus phone detection on the surface and temporal distribution of events indicates the suitability of this technique for colocated collaboration in small groups.

ACKNOWLEDGEMENTS

This work is supported by the NoE INTERMEDIA funded by the European Commission (NoE 038419).

REFERENCES

1. T. Cuypers, Y. Francken, C. Vanaken, F. Van Reeth, and P. Bekaert. Smartphone localization on interactive surfaces using the built-in camera. In *Proc. Procams*, 2009.
2. J.Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST*, pages 115–118, 2005.
3. R. Hardy and E. Rukzio. Touch & Interact: Touch-based interaction of mobile phones with displays. In *Proc. MobileHCI*, pages 245–254, 2008.
4. K. Hinckley. Synchronous gestures for multiple users and computers. In *Proc. UIST*, pages 149–158, 2003.
5. L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H. Gellersen. Smart-Its friends: A technique for users to easily establish connections between smart artefacts. In *Proc. UbiComp*, pages 116–122, 2001.
6. R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Trans. Mob. Comp.*, 8:792–806, 2009.
7. D.L. Mills. Internet time synchronization: The network time protocol. *IEEE Trans. Comm.*, 39:1482–1493, 1991.
8. J. Rekimoto. Pick-and-Drop: A direct manipulation technique for multiple computer environments. In *Proc. UIST*, pages 31–39, 1997.
9. J. Rekimoto. SyncTap: Synchronous user operation for spontaneous network connection. *Personal and Ubiquitous Computing*, 8:126–134, 2004.
10. S. Ronkainen, J. Häkkilä, S. Kaleva, A. Colley, and J. Linjama. Tap input as an embedded interaction method for mobile devices. In *Proc. TEI*, pages 263–270, 2007.
11. M. Strohbach, G. Kortuem, H. Gellersen, and C. Kray. Using cooperative artefacts as basis for activity recognition. In *Proc. EUSAI*, pages 49–60, 2004.
12. A. D. Wilson and R. Sarin. BlueTable: Connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. GI*, pages 119–125, 2007.