

# A Cross-Device Interaction Style for Mobiles and Surfaces

Dominik Schmidt<sup>1</sup>, Julian Seifert<sup>2</sup>, Enrico Rukzio<sup>2,1</sup>, Hans Gellersen<sup>1</sup>

<sup>1</sup>School of Computing and Communications, Lancaster University, UK

{schmidtd, hwg}@comp.lancs.ac.uk

<sup>2</sup>paluno, University of Duisburg-Essen, Germany

{enrico.rukzio, julian.seifert}@uni-due.de

## ABSTRACT

Natural forms of interaction have evolved for personal devices that we carry with us (*mobiles*) as well as for shared interactive displays around us (*surfaces*) but interaction across the two remains cumbersome in practice. We propose a novel cross-device interaction style for mobiles and surfaces that uses the mobile for tangible input on the surface in a stylus-like fashion. Building on the direct manipulation that we can perform on either device, it facilitates fluid and seamless interaction spanning across device boundaries. We provide a characterization of the combined interaction style in terms of input, output, and contextual attributes, and demonstrate its versatility by implementation of a range of novel interaction techniques for mobile devices on interactive surfaces.

## Author Keywords

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

## ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces: Input devices and strategies, Interaction Styles

## General Terms

Design, Human Factors

## INTRODUCTION

Mobiles and surfaces and their associated forms of interaction become ever more powerful and pervasive in our lives, but in their separate ways. By mobiles we mean small devices of a personal nature that we have with us for interaction; devices that are highly personalized, that store private data, and that are a proxy of ourselves in the digital world. By surfaces we mean larger displays that we walk up to for interaction, that let us interact with content at larger scale, that are more public and that afford sharing. In terms of interaction, rich and distinctive direct manipulation styles have evolved with either class: mobiles incorporate multimodal interfaces and can sense in 3D how they are being manipulated, and surfaces support multi-user and multi-touch interaction. However, in spite of the advances on either platform, it remains cumbersome in practice to interact across mobiles and surfaces.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DJS 2012, June 11-15, 2012, Newcastle, UK.

Copyright 2012 ACM 978-1-4503-1210-3/12/06...\$10.00.



Figure 1. A natural style for cross-device interaction with mobiles and surfaces. The mobile is used like a stylus for direct selection on the surface. For example, users can pick up content from the surface by touching it with their phone.

There are compelling reasons for combined use of mobiles and surfaces, and for seamless interaction across the two. Mobiles are great for carrying data and media while surfaces offer better scale for interaction with content. Mobiles provide user control over personal data while surfaces make it easy to share. Surfaces can be used by multiple users in the same way while mobiles can be used in highly personalized ways.

These are not new insights. Ever since the advent of mobiles, researchers have investigated topics such as coupling with surfaces for larger display [30], sharing of personal data on surfaces for collaboration [11], direct manipulation techniques for data transfer [29], remote interaction with larger surfaces [22], and private input, output and authentication around public devices [21, 6]. For any of these concerns, numerous techniques have been demonstrated. Data transfer, for instance, has been shown in terms of hyperdragging in augmented environments [29], “squirting” onto target devices in wireless networks [18], touching of target areas using NFC [14], tangible placement of mobiles on horizontal surfaces [37], as well as other ways. However, no single interaction style has emerged that would underpin mobile-surface-interaction more generically.

In this work we investigate a novel cross-device interaction style that we envision to be a generic platform for synergistic interaction with mobiles and surfaces across a variety of tasks and applications (Figure 1). The essence of this style is that the mobile device is used for selection of targets on a surface by direct touch, creating touch events that are associated with a position on the surface and the identity of the mobile. We build on the recently introduced *PhoneTouch* technique which demonstrated that this style can be implemented on multi-touch surfaces concurrently with finger touch sensing, by combining touch sensing embedded in the mobile with touch detection on the surface [32].

We investigate our vision of a generic interaction style for mobiles and surfaces in three steps. Our first step is to characterize

the interactions that are enabled by fusing mobile and surface. We identify the fundamental input, output, and contextual attributes that define the building blocks for mobile-surface interaction techniques. The second step is to demonstrate our vision by implementation of a range of interactions. We do this in the concrete setting of smartphone use on an interactive tabletop, building on top of the PhoneTouch method for sensing mobile touch events. For the purposes of our main argument, we show that our interaction style is generic and capable of underpinning a versatile range of interaction goals. At the same time we show that the way in which we facilitate integration of mobiles is effective in addressing practical challenges in surface interaction, by introducing novel techniques for data transfer, personalization, user interface composition, authentication, localized and private feedback, and input expressiveness. Our third step is to illustrate our interaction style in a number of applications we have built, to demonstrate the flow and fluidity of interaction across mobile and surface.

## RELATED WORK

How mobile devices and larger display surfaces can be used in complementary ways has been explored widely, in early visions [10, 29], work on combining PDAs with single display groupware [11, 22] and recent interest in the use of phones with interactive surfaces [37, 32]. Work has frequently been driven by specific application agendas, such as working across personal and shared contexts [11], whereas our concern is to generally support symbiotic use of small personal devices and large displays [26].

Many of the techniques that have been developed are for interaction at a distance and do not involve direct contact between mobile and surface. This includes indirect manipulation techniques, such as data synchronization through a standard user interface [11], remote cursor control by stylus input on a handheld [22], and mouse-like cursor control by relative motion of a mobile phone [20, 3]. Other techniques permit direct manipulation at a distance by pointing of a mobile with respect to the surface. This was first demonstrated with a PDA-mounted laser for direct pointing at a remote display, coupled with fine-grained interactions that then take place on the handheld screen, using a linked representation [23]. Recent work has leveraged built-in cameras, including Point & Shoot using camera-phones as view finders for remote selection on large displays [3], and related work by Pears *et al.* adding direct interaction on the phone's touch screen [25], and Touch projector with and additionally enables multi-touch interactions [4].

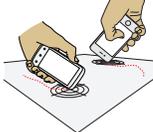
A range of recent works have contributed techniques that are based on contact between mobiles and surfaces, enabling initial device association in a physically grounded manner [37, 32]. Many of these techniques have been developed specifically for horizontal surfaces (tabletops), and involve placement of the mobile on the surface for the entire duration of the interaction [37, 24, 8, 15]. In the case of BlueTable the placement is used to initiate data transfer from mobile phones: as soon as a phone is placed on a surface, it transfers content (photos) for exploration onto the surface [37]. Microsoft Surface Mobile Connect is a related commercial system to share photos and other media [19]. In other work, mobiles are used more dynamically on the surface. Amnesia's Razorfish Connect [1],

for instance, allows users to freely move iPhones or iPads on a multi-touch tabletop as magic lenses or focus plus context screens [24].

The interaction style we propose builds on PhoneTouch, a 'lighter' touch technique for mobiles on surfaces, where the mobile is not placed on the surface but used like a stylus [32]. While adopting stylus-like interaction, PhoneTouch is fundamentally different from pen-based techniques (such as Pick-and-Drop [27]) as the primary concern is to leverage the mobile for symbiotic use with the surface. In related work, mobile phones were used for touch interaction with NFC-tagged displays, however with a coarser-grained touch resolution due to the tag size ( $4 \times 4 \text{ cm}$ ) [14]. Other work has employed similar touch techniques but not focused on interaction [16].

## INPUT & OUTPUT SPACE

We first consider the input and output attributes that characterize the proposed mobile-surface interaction style. The sequence in which we present the attributes does not imply any dependence. Each attribute stands by itself, and attributes can be combined in a variety of ways, as we will illustrate in subsequent sections (see Table 3).

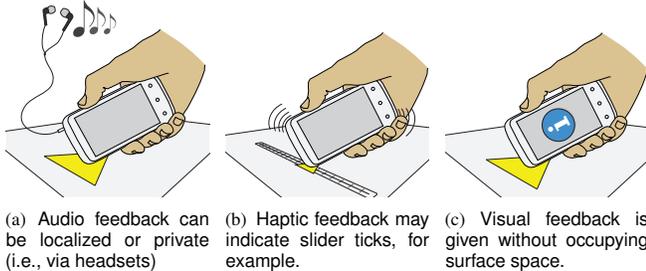
Source, Type	Input Attribute
Mobile, fixed value 	<b>Identifier</b> The mobile's identifier allows different mobiles to be distinguished. Therefore, each mobile touch can be associated to a corresponding mobile that is its source. Note that the identifier is fixed for a particular device.
Surface, 2D position 	<b>Location</b> The touch location is detected at the same granularity and within the same input space as finger touches. Note that finger and mobile touches are distinguished.
Mobile, 3D rotation 	<b>Orientation</b> The mobile's orientation determines the device part that is in contact with the surface (e.g., allowing different functions to be bound to each corner of the mobile). In addition, orientation can provide a stream of continuous input through rotation of the mobile around its axis while it touches the surface.

**Table 1. Basic input attributes associated with the touch of a mobile device on a surface.**

The user action at the heart of the proposed interaction style is a touch performed with a mobile on a surface. Table 1 describes the basic input attributes associated with this action: the identity of the mobile device, the location of the touch point on the surface, and the relative orientation of the mobile device during the touch. Table 2 captures additional input attributes that are available as a result of fusing the basic touch action with information on the mobile, input on the mobile, and input on the surface.

Source	Input Attribute
 <p>Mobile</p>	<p><b>Data Context</b> Mobiles (e.g., phones) hold a great amount of personal information which can provide useful data in the context of a touch action. For example, touching the recipient field in an email application on the surface could automatically fill in a selection of known contacts, using the address book on the mobile as source.</p>
 <p>Mobile</p>	<p><b>Selection</b> Users may explicitly select options to parameterize a touch. For example, they might first choose a command (e.g., “delete”) from a UI shown on the mobile, and then touch targets on the surface to apply it. Likewise, they could specify photos to transfer from mobile to surface. Input on the mobile during a touch could also trigger events to realize additional input states, similar to mouse clicks [5].</p>
 <p>Surface</p>	<p><b>Multi-touch</b> On the surface, a natural relationship between the touch with a mobile and finger touches from the user’s other hand is established based on proximity. This can be used for bimanual interactions. In particular, finger touches close to a mobile device touch point can be associated to the mobile and by extension to its user.</p>

**Table 2. Additional input attributes, which attach contextual information to a touch event.**



**Figure 2. Output on the phone provides localized or private feedback using multiple modalities.**

The output space for our interaction style is made up of the surface area under the touch, and output on the mobile. In contrast to feedback on the surface, output produced on the mobile can provide localized or private feedback. Visual, haptic, or audio responses may be given in the context of a touch by using output components commonly available on mobile devices, such as displays, vibrators, or speakers (Figure 2). For example, audio feedback can be localized using the mobile’s speaker, or private using a connected headset. Vibration can be used for unobtrusive feedback or alerts, and visual feedback on the phone mobile might be to provide tooltips or personalized information related to a touch.

### CROSS-DEVICE INTERACTION TECHNIQUES

For demonstration of our vision of mobile-surface interaction we have developed a demonstration system for smartphone interaction on an interactive surface. The implementation is based on Windows Phone OS 7 (WP7) on the phone side and .NET / Windows Presentation Framework (WPF) for the surface, and we have used HTC Mozart smartphones and custom-

built multi-touch surfaces for development of the interaction techniques that we introduce in this section, as well as for applications that we describe further below. The interactions we demonstrate are not dependent on this particular configuration, and we have for instance also used iPhones for demonstration of some of the cross-device techniques.

To detect and identify phone-touches, we use the synchronous PhoneTouch timing approach introduced by Schmidt et al. [32]. Both surface and phones sense touch events independently. The surface uses a camera, and phones their internal microphone. All detected events are time stamped and communicated over a wireless link. Valid phone-touches are identified based on event correlation in time. Although collisions are unavoidable, Schmidt et al. showed that this approach is suitable for collocated collaboration in small groups.

Our development of cross-device interaction techniques was framed by six issues in the use multi-touch surfaces that we identified as challenges that can be addressed in novel ways by symbiotic use of phone and surface:

- **Data Transfer.** Phones store personal information that are a rich data source for surface applications, and surfaces are natural for viewing of personal media [37]. Data transfer between phones and surface allows users to bring personal data to a larger display and facilitates sharing amongst users.
- **Personalization.** As their user’s proxy, phones can be used for personalization in shared interfaces to enable novel interactions that are otherwise not possible [31, 34].
- **User Interface Composition.** Command menus and tool palettes present challenges with respect to orientation, reachability, and clutter [35] which can be addressed by moving such interface elements from the surface to a mobile.
- **Authentication.** The inherently private process of authentication is a design challenge in shared interfaces [17] that can be addressed by integrating personal devices.
- **Localized & Private Feedback** Individual feedback can be beneficial in collaboration around shared surfaces [21, 13], for which phones can provide suitable channels.
- **Input Expressiveness.** Interactive surfaces enable natural input on a two-dimensional plane, but additional degrees of freedom can benefit certain tasks [2].

Table 3 provides an overview of the ten techniques we have conceived, to address the above issues. For each technique it is shown how it leverages attributes of the input & output space for mobile-surface interaction. All techniques exploit mobile device identity and touch point location as core feature, as well as selected other attributes. All techniques have been fully implemented with the exception of numbers 10 and 12, which represent more speculative ideas that we chose to only develop conceptually at this stage.

### Data Transfer

Data transfer across devices is naturally desirable around shared surfaces, for users to be able to bring personal data into a shared context and to collect data for personal use. We contribute two techniques to support this in a fast and fluid manner: *PhonePick&Drop* for transfer from phone to surface and vice versa and *PhoneCopy&Paste* for temporary transfer onto the phone as personal clipboard.

Issue	Technique	Identifier	Location	Orientation	Data Context	Selection	Multi-touch	Local Output*
Data Transfer	1 <b>PhonePick&amp;Drop</b> facilitates bidirectional data transfer between phones and surface.	✓	✓			✓		
	2 <b>PhoneCopy&amp;Paste</b> employs phones as personal clipboards for surface applications.	✓	✓			✓		
Personalization	3 <b>PhoneFill</b> makes existing personal information from phones available on the surface.	✓	✓		✓			
	4 <b>PhoneLenses</b> provide flexible ad hoc personalization of any surface content.	✓	✓				✓	
User Interface Composition	5 <b>PhonePalettes</b> move tool palettes and menus from surface to phone.	✓	✓			✓		
	6 <b>PhoneFaçades</b> afford fluid interface customization.	✓	✓			✓		
Authentication	7 <b>PhoneKey</b> enables fine-grained ad hoc authentication on the surface.	✓	✓		✓			
	8 <b>PhonePass</b> allows users entering passwords unobserved on their phone.	✓	✓			✓		
Localized & Private Feedback	9 <b>PhoneSpeaker</b> provides localized or private audio feedback.	✓	✓					✓
	10 <b>PhoneZone</b> <sup>†</sup> opens continuous private spaces spanning surface and phone.	✓	✓			✓	✓	✓
Input Expressiveness	11 <b>PhoneHandle</b> allows for direct manipulation of scalar controls.	✓	✓	✓				
	12 <b>PhoneGestures</b> <sup>†</sup> enables motion-based gestural input.	✓	✓	✓				

**Table 3.** Interaction techniques, the issue that they address, and input or output attributes that they use.

\*Local output indicates that audio, haptic, visual, or any combination of these three is conceptually required.

<sup>†</sup> *PhoneZone* and *PhoneGestures* have been developed conceptually; all other techniques have been fully implemented.

### 1 *PhonePick&Drop*

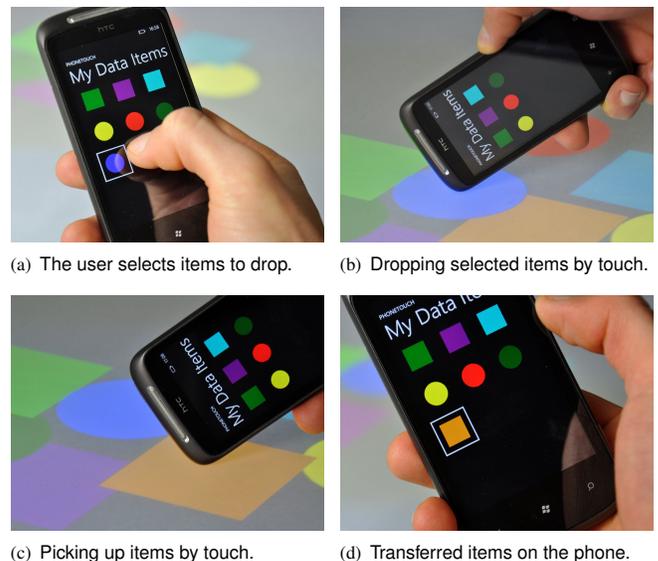
This technique is for intuitive transfer of data objects from phone to surface and vice versa. To transfer objects from phone to surface (drop), the user makes a selection in private on the phone (Figure 3(a)) and then touches the surface with the phone to initiate transfer and display at the selected location on the surface (Figure 3(b)). To transfer objects from the surface to the phone (pick), the user directly selects objects on the surface by phone-touch to initiate their transfer (Figure 3(c)). The technique inherits the simplicity of Rekimoto’s Pick-and-Drop [27] but is adapted for fast and intuitive data transfer between phone and surface. Users can in one step choose and transfer items to pick, and likewise select a target location and initiate a drop in one go.

*PhonePick&Drop* is based on a blackboard metaphor for sharing. This affords users to asynchronously interchange information from their phones via the surface, and to inspect items on the surface before picking them up onto their personal devices. In this respect, the technique is related to other work on data transfer by placement of phones on a shared surface [1, 19, 37]. In contrast to those works, our technique provides fine-grained control over which items to reveal: users select data items in private on the phone before dropping them onto the shared display. We believe this is important, as phones are very private devices that users would not typically share. For example, while it is common that users show content on their phones to others, they typically do so without giving their phone out of their hands.

### 2 *PhoneCopy&Paste*

This technique extends surface interaction with phones as personal clipboard. It is based on the same user actions as above for selection of objects and target locations on the

surface, however with a copy-and-paste semantic. *PhonePick&Drop* was motivated for sharing of phone content, while *PhoneCopy&Paste* is designed for manipulation of surface content, with phones serving as transient storage only. The enables multiple users to each have their individual clipboards, addressing the problem of correctly associating copy-and-paste actions in multi-user environments, and also affords visualization and efficient access to a history of copied items without consuming surface space.



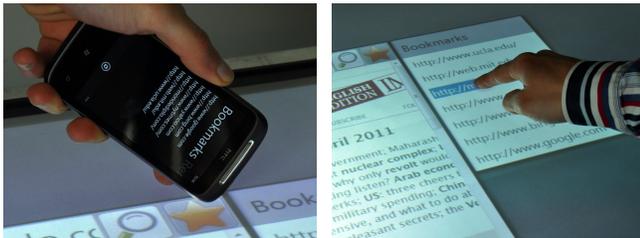
**Figure 3.** *PhonePick&Drop* allows users to transfer data between phone and surface in both directions.

## Personalization

Personalized interactions can be beneficial to multi-user applications [31, 34] but only few interactive surfaces support user identification directly (e.g., [7, 33]). Integration with phones offers an alternative by taking a proxy-based approach. Since each phone-touch is associated with a phone identifier, its user can be inferred. Phones also hold personal data of potential use for personalization of surface applications (e.g., user preferences). We introduce two techniques, *PhoneFill* and *PhoneLenses*, that leverage this potential.

### 3 PhoneFill

Users commonly store a variety of personal collections on their phone, such as contacts, music playlists, and browser bookmarks. This information can also benefit surface applications. For example, browsing the web on the surface, users may want to access a site they have previously bookmarked on their personal device, but manual transfer of the URL to the surface browser is tedious. To address this, we have designed the PhoneFill technique, enabling users to make existing personal information instantly available on the surface. This is illustrated in Figure 4: the surface object touched with the phone determines the context for PhoneFill. Based on this context, the phone identifies and provides relevant information to the surface application. In the shown example, the user phone-touches the browser’s bookmark control (Figure 4(a)). This triggers the phone to automatically retrieve and send its bookmark collection to the surface, where it becomes instantly available. Users can then choose from their personal bookmarks directly on the surface (Figure 4(b)), even in collaborative scenarios where browser interaction may be shared between multiple users. In the same way, a phone can provide contact details to send email from a surface application, or automatically fill-in payment forms.



(a) Touching a web browser... (b) ... to transfer personal bookmarks.  
**Figure 4.** *PhoneFill* makes existing personal information from the phone available to surface applications.

### 4 PhoneLenses

A phone-touch can act as a proxy for user identity, but is limited to a single point of contact. To extend user identification to multi-touch, we have integrated PhoneTouch with IdLenses, a concept for interaction through personalized lenses on a shared surface [34]. A user can open a PhoneLens by performing a phone-touch anywhere within an enabled surface application. The lens then defines an area in which any finger touch input is associated with the phone’s user. The lens moves along with the phone and disappears once the phone is lifted off the surface. The concept allows for any form of personalization of the surface content under the lens. Figure 5 shows an example from one of the applications we have built. A user explores



(a) Any finger touch through a lens is associated to the phone’s user. (b) The surface area under a lens is personalized (e.g., translated).

**Figure 5.** *PhoneLenses* can be invoked anywhere on the surface and move along with the phone.

the map of a museum with a personal lens through which content is adapted to their language, based on user preferences automatically provided by the user’s phone. In this way, a user can view content adapted to their needs, without disrupting other users viewing other parts of the shared display. Users can also provide input through the lens, for example marking a museum room on the shared map as a personal favorite.

## User Interface Composition

Placement of command menus and toolbars can be problematic on shared surfaces, as it can be difficult to make them easily accessible for different users in terms of physical reach and orientation [35]. We propose two techniques that address this issue with user interface composition across surface and phones. *PhonePalettes* are for off-loading of tool palettes to the phone, and *PhoneFaçades* support ad hoc customization of the interface.

### 5 PhonePalettes

The principal idea is to move tool palettes from the surface onto the phone to be close to hand on whichever part of a larger surface the user is working. This is related to the concept of detached user interfaces, in which tools were moved off the main display and onto a handheld device, in analogy to a painter whose focus lies on the canvas while keeping tools handy on a palette [28, 12]. However, PhonePalettes are different as the phone itself is used to select the target of a command on the surface. For demonstration of the technique, we have implemented a simplified graphics editor. In the example shown in Figure 6, the user selects the “Circle” command on the phone (Figure 6(a)) and then touches the surface to apply the command at the selected location (Figure 6(b)). The same command can be applied repeatedly on the surface. The commands can also be parametrized through the phone’s interface, for example to choose from different colors (Figure 6(c)). It is possible to preselect multiple compatible commands to be executed simultaneously with the next phone-touch (e.g., fill and stroke color can be applied together). Frequently applied tool settings can also be stored on the phone as personal preferences.

Input sequences that require prior selection of a mode (e.g., selecting “bold” before text entry) are common in graphical user interfaces, but problematic on interactive surfaces that lack the ability to distinguish touches of different users [31]. A related problem is visual feedback that indicates the selected mode [9]. PhonePalettes solve this issue as commands are selected on the user’s personal device, enabling multiple users



(a) Selecting tools and commands... (b) ... which are applied by touch.



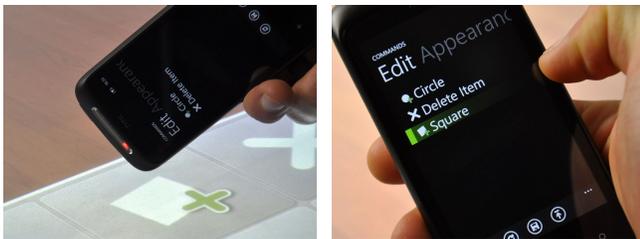
(c) PhonePalettes also support parameterized commands.

**Figure 6.** PhonePalettes move tool palettes and menus from surface to phone.

to work on a shared interface, each within their own mode. It is also possible to keep an audit trail based on the phone-touch identifier, and to provide per-user undo operations.

### 6 PhoneFaçades

Many tasks involve only a small set of an application’s command set. The idea of PhoneFaçades, inspired by User Interface Façades [36], is to enable users to pick commands from the surface in order to assemble a customized interface on the phone. In the example shown in Figure 7, the user picks the application’s “Square” command by selecting it with a phone-touch (Figure 7(a)). As a result, a representation of the command is automatically added to the phone (Figure 7(b)). The command is now ready to be used as described above for PhonePalettes. As this shows, PhoneFaçades involve only minimal overhead for interface customization. Users can pick up and arrange commands on their phone in an ad hoc fashion, to match their workflow. Multiple users can each assemble individually customized interfaces to use in a shared surface application.



(a) Picking up surface commands. (b) Customized command set.

**Figure 7.** PhoneFaçades lets users assemble a set of commands picked from the surface.

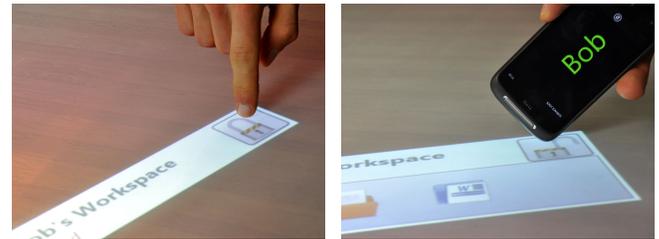
### Authentication

Authentication is an inherently private process that presents a distinct design challenge on shared surfaces. System-wide authentication is typically not appropriate due to the multi-user

context, but interaction-based authentication can be desirable, for example if users have different roles with varying levels of authority. We contribute two new techniques for authentication, *PhoneKey* for locking and unlocking of restricted content, and *PhonePass* for remote and unobserved password entry.

### 7 PhoneKey

PhoneKey is a technique for token-based authentication. Metaphorically, the phone serves as a key to control access to items on the shared surface, for example interactive applications or restricted content. To demonstrate the PhoneKey technique, we implemented multiple personal workspaces on a shared surface. A user can lock their workspace to protect enclosed data by touching the lock button (Figure 8(a)). To regain access, the user unlocks the workspace with a phone-touch on the lock button (Figure 8(b)). Authentication takes place implicitly, with the phone serving as the access token. Finger touches and other phones cannot unlock the workspace, but could for instance trigger display of an alert message.

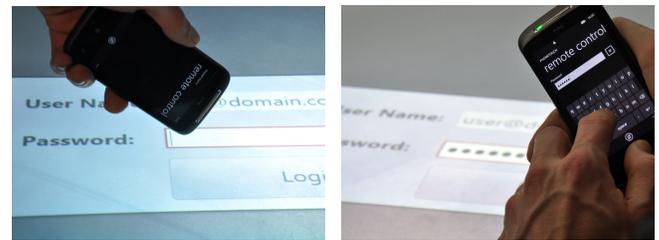


(a) Locking a workspace.

(b) Access with an authorized phone.

**Figure 8.** PhoneKey enables token-based authentication and fine-grained access control.

PhoneKey facilitates lightweight and fine-grained access control. Any surface element that is accessible by finger touch is also accessible by phone-touch, and so can be protected using PhoneKey. The technique is generic and not bound to any particular underlying authentication mechanism. A unique device ID may be sufficient for some applications, but depending on required security, PhoneKey could also be combined with cryptographic key exchange between phone and surface.



(a) The user touches a password field on the surface to select it...

(b) ... and enters the password unobserved via the phone.

**Figure 9.** PhonePass enables users to enter passwords unobserved on their phone.

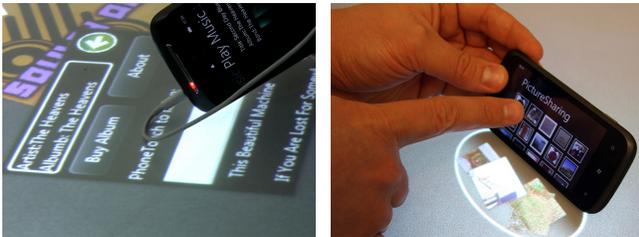
### 8 PhonePass

Many existing applications, for example many web-based services, require users to enter a password to log in. This is problematic on shared services due to potential shoulder-surfing attacks [17]. PhonePass addresses this problem by

enabling users to enter passwords via their phone. The user initiates this interaction by touching the password field with their phone (Figure 9(a)). A corresponding control appears on the phone where the password can be entered unobserved (Figure 9(b)). The only feedback on the surface is given in form of disguised characters, shown as asterisks. The phone could also store a collection of passwords in an internal vault. A phone-touch on a password field would then automatically retrieve and fill-in the matching password, similar to a PhoneFill interaction. In contrast to other proposed surface authentication approaches [17], PhonePass integrates readily with existing applications. Closely related is PocketPIN [6], but PhonePass affords a simpler method of associating phone and password field.

### Localized & Private Feedback

Output on large displays and shared surfaces is public by default. However it has been argued that private feedback can be beneficial (e.g., individual audio channels have led to a more equitable task participation [21]). We contribute two new feedback techniques: *PhoneSpeaker* provides a personal audio channel, and *PhoneZone* an output zone that is shielded from the view of other users.



(a) Using *PhoneSpeaker* for private audio feedback. (b) *PhoneZone* creates a private space across phone and surface.

Figure 10. Localized and private feedback.

#### 9 PhoneSpeaker

This technique implements audio feedback for phone-touch events. Localized feedback is given through the phone's internal speaker, and private feedback when headphones are connected. For example, localized feedback of affirmative or negative sound has been shown to raise user awareness of input errors [13]. Private audio feedback can be useful to access audio content on surfaces while avoiding interference among multiple users. In an example application that we have implemented, multiple users can browse a music collection on a shared surface and preview tracks individually by touching them with their phones (Figure 10).

#### 10 PhoneZone

The idea of *PhoneZone*, developed as concept but not fully implemented, is to provide the user with a visual output area that is shielded from other users. As shown in Figure 10(b), the concept is for users to place their phone sideways on the surface in order to open a private space that combines display space on the phone with display space on the surface, "in the shadow of the phone". Like the horizontal hand gesture proposed by Wu et. al [38], the phone shields parts of the surface, thereby blocking it from other users' view. The combined display space of a *PhoneZone* could also be exploited

for direct manipulations, such as sliding content off the phone down onto the surface.

### Input Expressiveness

Input on interactive surfaces is inherently two-dimensional. We contribute two new techniques that exploit the phone as a device that we use for planar interaction on the surface but that offers additional degrees of freedom (in similar ways as the Rockin' Mouse [2]). *PhoneHandle* uses device motion to manipulate scalar controls, and *PhoneGestures* enables discrete gestures performed with the phone.

#### 11 PhoneHandle

The idea of this technique is to manipulate scalar controls that are selected by a phone-touch on the surface. As the phone is held like a stylus with a contact point on the surface, it can be manipulated naturally in terms of varying pitch (forward-backward tilt), yaw (left-right tilt), or roll (rotation around the z-axis). The range of device motions is constrained by the way a phone is held, and limitations of arm and wrist movement, but this can be accommodated in the design of the control. Figure 11(a) shows an example of a slider control that we have implemented as *PhoneHandle*. The slider on the surface is selected by a phone-touch, and the user manipulates its value by tilting the phone forward and backward, using a rate control mapping. A second example, in Figure 11(b) shows rotation of a phone to control a knob displayed on the surface.



(a) Slider control by tilting.

(b) Knob control by rotation.

Figure 11. *PhoneHandle* allows users to directly manipulate scalar controls through device motion.

#### 12 PhoneGestures

*PhoneGestures* are based on the same device motions used in *PhoneHandle*, but afford discrete gestures instead of direct motion-to-control mapping. We have not included *PhoneGestures* in our current implementation, but conceptually we foresee two types of gestures. The first type are counter-intuitive movements of a phone in extension of a phone-touch, as a safety catch to avoid accidental activation of critical functions. For example, in order to delete items on the surface permanently, a user could be required to not only select the delete button but to also perform a rotation of the phone while the button is selected. The interaction metaphor here is that of a launch key. Such a gesture can be designed to be unlikely to be performed accidentally but to be easily integrated into the touch interaction flow.

The second type of gesture we propose are metaphorical. For example, a phone-touch could be combined with screwdriver motion, for fastening or unfastening of an object on the surface. It could also be combined with a pumping motion, to lift

or lower an object selected on the surface. This allows for additional forms of expression that may build on metaphors that are intuitive in particular application contexts, or that provide for playful interaction in games and entertainment.

### APPLICATIONS

We have implemented a number of applications to further illustrate mobile-surface interaction. The applications make use of a variety of the interaction techniques and demonstrate them in realistic flows of interaction. They also highlight the fluidity of the interaction style in terms of seamlessly moving between interactions on the phone and on the surface.



(a) Users arrange words in private on their phone to then... (b) ... drop them onto the shared word game board.

**Figure 12.** *Word game* illustrates seamless integration of private and shared interactions.

### Word Game

This application is a clone of the Scrabble board game. It makes use of PhonePick&Drop and illustrates use of the mobile device for private display and interaction fluidly interwoven with shared interaction on the surface. Between two and four players individually form words to then place them into a shared grid like a crossword. In our implementation, players receive a set of letters on their phone to arrange them in private (Figure 12(a)). Once it is a player's turn, the word is dropped by a phone-touch on the surface, simultaneously specifying the target cells on the displayed game board (Figure 12(b)). Players can also pick up and rearrange letters using PhoneCopy&Paste if their initial choice of word does not fit as planned.



**Figure 13.** The *collage designer* implements PhonePalettes, letting users apply various settings, such as changing a photo's frame.

### Collage Designer

This is an application for multiple users to bring their photos to the surface, and to creatively arrange them into a combined collage, making use of PhonePalettes in addition to the techniques demonstrated in the word game. Users can freely move, rotate, and scale photos with multi-touch, and they can use editing tools, for instance to choose from different picture

frames, to add captions, or to delete photos. Editing options were integrated in two ways, on PhonePalettes as shown in Figure 13, and alternatively via context menus on the surface. This provides users with choices in their workflow.

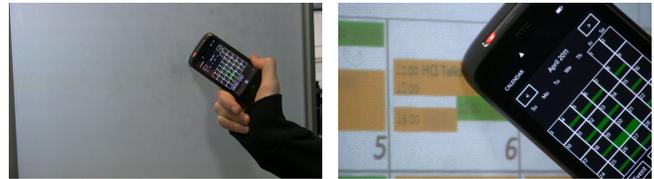


(a) Browsing music using multi-touch. (b) Using the phone to purchase.

**Figure 14.** *Music store* lets users browse music manually and integrates a phone for transactions.

### Music Store

The music store application allows users to browse through different albums using common multi-touch interaction on a shared surface (Figure 14(a)). To preview a song we apply PhoneSpeaker, providing individual audio over headphones. In doing so, multiple users can listen to songs without disturbing each other. Users can directly buy songs by touching them with their phone, thus using the phone not only to transfer the music to, but also for authenticating the purchase (Figure 14(b)).



(a) Dropping a calendar for sharing. (b) Unlocking event detail.

**Figure 15.** *Calendar* lets users share their personal calendars with fine-grained privacy control.

### Calendar

Personal calendars are a standard application on mobile phones. However, scheduling or sharing events amongst multiple co-located users is not readily supported, as each user has access to their own calendar only. We address this with a calendar sharing application that we built on our platform. The application lets users drop their personal calendar onto a surface by tapping the surface while the application is open on the phone (Figure 15(a)). The calendar becomes shared in a privacy-sensitive way, initially only showing the blocked times but without event details. This allows users to jointly look for free slots. The owner of the calendar can also selectively unlock individual entries in the calendar using PhoneKey. This makes the event detail visible, and allows other users to copy the event to their own calendars, using PhonePick&Drop (Figure 15(b)).

### Exposure to Users

We recruited six graduate students (2 female, 4 male; aged 23 to 29) for trial sessions to expose the techniques we developed to use by others but ourselves. Five of them were smartphone users (4 iPhone, 1 Nexus One), one owned a Nokia phone without touch screen. The participants individually tried the ten techniques we had fully implemented, and

were paired up for multi-user trial of the word game and the collage designer. Each trial session lasted about 30 minutes. We observed the participants' behavior and comments, and conducted informal post-study interviews about their opinions and suggestions.

Participants found the style of interaction natural. One participant suggested that users might be concerned about using expensive phones for input by impact on a surface, but none of the participants showed any hesitation in applying the phone-touch techniques. Of all techniques, PhonePass and PhoneFill appealed most to the participants as they address problems that users found familiar and important, beyond the setting of interactive tabletops that was used in the trial session.

In the multi-user part of the trial, users engaged quickly with the word game as its board game variant is well-known. Without having been prompted to do so, they intuitively used the phone as private screen, and naturally moved back and forth between interaction on the phone and on the surface. The collage designer presented a more artificial task but it prompted participants to comment on the distinct ease by which photos stored on the phone can be shared for joint viewing on a larger screen, a process that is clearly perceived as cumbersome with state of the art camera phones and digital cameras. The PhonePalette technique was learned without difficulties, but some participants had reservations about its value as the same options were accessible also via the surface. Yet several participants perceived PhonePalettes as a shortcut and possible solution to orientation problems of menus on tabletops.

Participant also made suggestions for improvement. In particular, they proposed to clearly indicate which surface functions can be accessed via phone versus finger touches. Further, several participants suggested that the phone could be used "like a finger for the other hand", indicating that the phone was seen as unobtrusive tool.

## DISCUSSION

The introduced interaction style is suitable for any settings in which users interact directly with touch-sensitive surfaces. Most of the shown examples were based around a tabletop, but the interaction style and introduced techniques can be equally applied on vertical surfaces. Some of the techniques could also be considered for interaction with other devices than interactive displays. For example, a user could drop a photo onto a printer, by phone-touching the printer's UI.

The concept is generic and versatile as demonstrated by the diversity of techniques and applications we have described above. As a direct manipulation style, the concept is well suited for interactions that take place in the context of specific elements or objects on an interactive surface, where fine-grained selection is key. It lends itself to applications that involve a combination of interaction on mobiles and interaction on surfaces, where people need to easily switch between the devices. Touch input with a mobile as opposed to pen or finger is useful when additional features of the mobile are exploited; this may simply be the identifier, but can also be stored data or user-selected options in the mobile UI.

The interactions are consistently based on direct manipulation (on the phone, on the surface, and of the phone on the surface)

to promote a natural interface. However, users might find it less natural to use a mobile as a stylus, and they might be concerned about causing damage to either their mobile or the surface, as both represent expensive devices. In our user trial sessions this did not appear to be an issue, however participants were provided with devices and might have been concerned if they had been asked to use their own. The concern can be addressed with inexpensive bumpers or covers that are already widely used.

The implementation of our concept is based on a distributed sensing approach. Previous work has shown that the temporal correlation of impact sensing in the phone and touch recognition on the surface leads to robust detection[32]. Device association with a touch can only be confused if it occurs in the same recognition frame (approx. 25ms). Depending on application, simultaneous touch might be ruled out (e.g., *word game*) or is not critical. In other cases it might be appropriate to integrate a recovery mechanism, such as undo, or to consider a different sensing approach that does not depend on temporal correlation only. For instance, both mobile and surface could measure impact and correlate impact patterns to discriminate simultaneous touch.

The interaction style is new but it does not require new or specialized hardware. On the mobile side, we used unmodified off-the-shelf components only in our implementation. On the surface side, we used a standard setup for vision-based multi-touch that did not require any hardware modification for our purposes. We would expect that detection of mobile device touch can also be integrated with alternative surface technologies, in inexpensive ways.

## CONCLUSION

The vision underlying this work was to support natural interaction across personal mobile devices, and larger interactive surfaces. Our contribution is an interaction style that fuses direct manipulation interactions of three kinds: interactions we perform on mobiles, multi-touch interactions on surfaces, and stylus-like use of mobiles on a surface. We have shown that this interaction style readily gives rise to a versatile range of interaction techniques and application concepts. Up to this point, we have covered a breadth of interaction techniques, rather than providing in-depth analysis. We believe, however, that additional work into this direction is clearly beneficial. Promising directions for future work also include using multiple surfaces at a time, integrating interactions into further application domains, and long-term deployments of our techniques in a real-world setting, to gain a deeper understanding of their use.

## ACKNOWLEDGEMENTS

Parts of this work were conducted within the Emmy Noether research group Mobile Interaction with Pervasive User Interfaces funded by the German Research Foundation (DFG). We would further like to thank Sascha Geeren for implementation support and fruitful discussions, Paul Holleis for valuable feedback, and all our study participants.

## REFERENCES

1. Amnesia. Razorfish Connect. 02 March 2011.  
<http://www.amnesiarazorfish.com.au/2011/01/>

2. R. Balakrishnan, T. Baudel, G. Kurtenbach, and G. Fitzmaurice. The Rockin' Mouse: Integral 3D manipulation on a plane. In *Proc. CHI*, pages 311–318, 1997.
3. R. Ballagas, M. Rohs, and J. Borchers. Sweep and Point & Shoot: Phocam-based interactions for large public displays. In *CHI Ext. Abstracts*, pages 1200–1203, 2005.
4. S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: Mobile interaction through video. In *Proc. CHI*, pages 2287–2296, 2010.
5. W. Buxton. A three-state model of graphical input. In *Proc. INTERACT*, pages 449–456, 1990.
6. A. De Luca and B. Frauendienst. A privacy-respectful input method for public terminals. In *Proc. NordiCHI*, pages 455–458, 2008.
7. P. H. Dietz and D. Leigh. DiamondTouch: A multi-user touch technology. In *Proc. UIST*, pages 219–226, 2001.
8. F. Echtler, S. Nestler, A. Dippon, and G. Klinker. Supporting casual interactions between board games on public tabletop displays and mobile devices. *Pers. and Ubiqu. Comp.*, 13:609–617, 2009.
9. K. Everitt, C. Shen, K. Ryall, and C. Forlines. Modal spaces: spatial multiplexing to mediate direct-touch input on large displays. In *CHI Ext. Abstracts*, pages 1359–1362, 2005.
10. G. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Comm. ACM*, 36:39–49, 1993.
11. S. Greenberg, M. Boyle, and J. Laberge. PDAs and shared public displays: Making personal information public, and public information personal. *Pers. and Ubiqu. Comp.*, 3:54–64, 1999.
12. D. Grolaux, J. Vanderdonck, and P. Van Roy. Attach me, detach me, assemble me like you work. In *Proc. INTERACT*, pages 198–212, 2005.
13. M.S. Hancock, C. Shen, C. Forlines, and K. Ryall. Exploring non-speech auditory feedback at an interactive multi-user tabletop. In *Proc. GI*, pages 41–50, 2005.
14. R. Hardy and E. Rukzio. Touch & Interact: Touch-based interaction of mobile phones with displays. In *Proc. MobileHCI*, pages 245–254, 2008.
15. T. Hesselmann, N. Henze, and S. Boll. FlashLight: Optical communication between mobile phones and interactive tabletops. In *Proc. ITS*, pages 135–138, 2010.
16. W. Hutama, P. Song, F. Chi-Wing, and W. B. Goh. Distinguishing multiple smart-phone interactions on a multi-touch wall display using tilt correlation. In *Proc. CHI*, 2011.
17. D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proc. CHI*, pages 1093–1102, 2010.
18. T. et al. Kindberg. People, places, things: Web presence for the real world. *Mob. Netw. Appl.*, 7(5):365–376, 2002.
19. Microsoft. Surface Mobile Connect. 24 March 2012. <http://technet.microsoft.com/en-us/library/ee692087.aspx>.
20. K. Miyaoku, S. Higashino, and Y. Tonomura. C-Blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal. In *Proc. UIST*, pages 147–156, 2004.
21. M.R. Morris, D. Morris, and T. Winograd. Individual audio channels with single display groupware: effects on communication and task strategy. In *Proc. CSCW*, pages 242–251, 2004.
22. B. A. Myers. Using handhelds and PCs together. *Comm. ACM*, 44:34–41, 2001.
23. B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Hong, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: Measuring the performance of laser pointers and other devices. In *Proc. CHI*, pages 33–40, 2002.
24. A. Olwal and S. Feiner. Spatially aware handhelds for high-precision tangible interaction with large displays. In *Proc. TEI*, pages 181–188, 2009.
25. N. Pears, D.G. Jackson, and P. Oliver. Smart phone interactions with registered displays. *IEEE Perv. Comp.*, 8:14–21, 2009.
26. M. Raghunath, C. Narayanaswami, and C. Pinhanez. Fostering a symbiotic handheld environment. *IEEE Comp.*, 36:56–65, 2003.
27. J. Rekimoto. Pick-and-Drop: A direct manipulation technique for multiple computer environments. In *Proc. UIST*, pages 31–39, 1997.
28. J. Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *Proc. CHI*, pages 344–351, 1998.
29. J. Rekimoto and M. Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proc. CHI*, pages 378–385, 1999.
30. S. Robertson, C. Wharton, C. Ashworth, and M. Franzke. Dual device user interface design: Pdas and interactive television. In *Proc. CHI*, pages 79–86, 1996.
31. K. Ryall, A. Esenther, K. Everitt, C. Forlines, M. R. Morris, C. Shen, S. Shipman, and F. Vernier. iDwidgets: parameterizing widgets by user identity. In *Proc. INTERACT*, pages 1124–1128, 2005.
32. D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen. PhoneTouch: A technique for direct phone interaction on surfaces. In *Proc. UIST*, pages 13–16, 2010.
33. D. Schmidt, M. Chong, and H. Gellersen. HandsDown: Hand-contour-based user identification for interactive surfaces. In *Proc. NordiCHI*, pages 432–441, 2010.
34. D. Schmidt, M. Chong, and H. Gellersen. IdLenses: Dynamic personal areas on shared surfaces. In *Proc. ITS*, pages 131–134, 2010.
35. C. Shen, K. Ryall, C. Forlines, A. Esenther, F. Vernier, K. Everitt, M. Wu, D. Wigdor, M.R. Morris, M. Hancock, and E. Tse. Informing the design of direct-touch tabletops. *IEEE Comp. Graph. and App.*, 26:36–46, 2006.
36. W. Stuerzlinger, O. Chapuis, D. Phillips, and N. Roussel. User interface façades: Towards fully adaptable user interfaces. In *Proc. UIST*, pages 309–318, 2006.
37. 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.
38. M. Wu and R. Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proc. UIST*, pages 193–202, 2003.