

PointerPhone: Using Mobile Phones for Direct Pointing Interactions with Remote Displays

Julian Seifert¹, Andreas Bayer², and Enrico Rukzio¹

¹ Ulm University, Institute of Media Informatics,
James-Franck-Ring, 89081 Ulm, Germany

² University of Duisburg-Essen, Schützenbahn 70, 45127 Essen, Germany
{julian.seifert, enrico.rukzio}@uni-ulm.de,
andreas.bayer@uni-due.de

Abstract. Large screens or projections in public and private settings have become part of our daily lives, as they enable the collaboration and presentation of information in many diverse ways. When discussing the shown information with other persons, we often point to a displayed object with our index finger or a laser pointer in order to talk about it. Although mobile phone-based interactions with remote screens have been investigated intensively in the last decade, none of them considered such direct pointing interactions for application in everyday tasks. In this paper, we present the concept and design space of PointerPhone which enables users to directly point at objects on a remote screen with their mobile phone and interact with them in a natural and seamless way. We detail the design space and distinguish three categories of interactions including low-level interactions using the mobile phone as a precise and fast pointing device, as well as an input and output device. We detail the category of widget-level interactions. Further, we demonstrate versatile high-level interaction techniques and show their application in a collaborative presentation scenario. Based on the results of a qualitative study, we provide design implications for application designs.

Keywords: Mobile phone, pointing, interaction, collaboration.

1 Introduction

Large screens and projections have become part of our daily lives. They support collaboration as they allow multiple users to simultaneously access information. For instance, in a meeting presentation, information is shared with multiple users. However, the control of what is being displayed is usually limited to a single user (e.g., in presentation). Others cannot share, access, or manipulate virtual objects or data on the remote display.

Direct pointing has been investigated for interaction with remote displays [19]. Direct pointing interaction is a natural way for users to select and interact with objects on a remote screen (see Figure 1(a)) [16]. However, such settings enable only few options for interactions and are limited to basic operations.

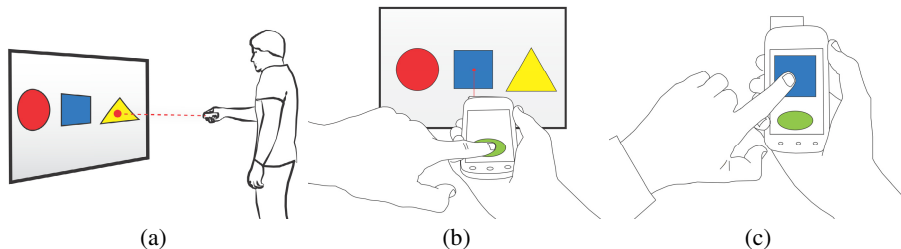


Fig. 1. Using the mobile phone as a pointing device enables versatile interactions with remote screens: (a) Pointing to targets. (b) Performing actions on the personal phone such as selecting and downloading an item. (c) Further interaction with data on the phone.

As mobile phones are ubiquitously available, they enable users to access remote displays in diverse ways [2]. For instance, downloading information from a remote display to the mobile phone for further inspection [4] or sharing information on a remote screen with others [1]. Pointing-based interactions offer an easy-to-use way of allowing users to interact with an object by pointing to it [24].

Hence, mobile phones with integrated pointing abilities enable diverse novel options for interaction in a natural and seamless way. Using the mobile phone as a pointing device and general interaction device has not been investigated, thus raising questions regarding how the phone’s specific characteristics (e.g., options for input and output) and attributes (e.g., user data context) can be integrated into the interaction process.

In this paper, we contribute the detailed investigation of the novel design space of PointerPhone. PointerPhone uses mobile phones as pointing devices for new direct and natural pointing interactions with remote screens (see Figure 1). We present a classification consisting of low-level, widget-level, and high-level interaction techniques. Further, we show application examples and demonstrate the integration of diverse techniques into a collaborative meeting support application which we implemented based on a prototype system that uses laser-pointer-equipped mobile phones¹. Furthermore, we present observations and results of a qualitative user study and we provide a catalogue of design guidelines as well as lessons learned that need to be considered when designing applications based on PointerPhone interactions.

2 Related Work

A large body of research exists in the area of remote interaction with remote information displays such as large screens or projections. These related works can be classified into the following categories: a) general pointing interaction and technologies, b) interaction with remote screens using mobile phones as a pointing device, and c) collaboration support. The review of the related work shows that no previous work investigated the combination of mobile phones with direct pointing for interaction with remote screens.

¹ A complementary video figure is available at <http://youtu.be/qp3pIk1YLxo>

2.1 Pointing Interaction

Early work by Rekimoto investigated the potential of interacting with remote screens in a continuous work space [23]. Using the hyperdragging technique, users could move data items across multiple devices (e.g., laptops or shared screens). Here the user controls a cursor with a mouse connected to one device. The cursor would *jump* to other screens when reaching a display border, allowing to reach out for data across devices. By using a mouse, the pointing is only indirect as users cannot literally point out to the desired target but rather move the mouse on a planar surface. Olsen and Nielsen investigated direct pointing for interaction with virtual objects based on laser pointers [19]. They analyzed interaction techniques based on camera tracking of a simple laser pointer on a remote screen (e.g., interaction with widgets, drawing, and navigation). Vogt et al. demonstrated how collaborative multi-user settings can be implemented using several laser pointers [31]. Myers et al. compared the effect of different form factors of pointing devices (a pen-like pointer, a pointer attached to a toy gun, and a pointer attached to a personal digital assistant device) for interaction with large projected screens [16]. They observed that it is essential to support interactions which do not cause much jittering of the pointing device and allow users to easily control the making of selections. Shizuki et al. investigated the potential of using pointers to interact with remote screens beyond making mouse-like selections [27]. Their findings show that simple commands for applications can be directly performed using a standard laser pointer device. Nintendo released the Wii U controller device which enables pointing to a remote screen and interaction through a small touchscreen [18]. Similarly to the PointerPhone approach, this system enables interaction on a remote and a personal screen. Yet, the Wii U game controller does not provide and store users' data such as photos, bookmarks or contacts. Also, a game controller is not always carried by the users and thus is not suited for interacting in diverse settings. Schmidt et al. introduced a pico projector-based approach where users project an interface onto smart objects that are equipped with a specific sensor [25]. Through moving the projector and thus the projected interface, users can select and activate different interface widgets. This way of pointing interaction is, however, limited to one user and does not support collaboration.

2.2 Pointing Technologies

A large variety of technologies and approaches exist to allow for the sensing of the pointing interaction. They can be classified into direct pointing (e.g., with laser pointers as in [19, 31, 29]), and inertial sensing (e.g., using gyroscope and accelerometer sensor data as in [18]). An additional approach was presented by Zizka et al.; they presented SpeckleSense which enables precise interactions with remote screens based on laser speckle motion sensing [34]. In contrast to a pointer-based approach, each interaction device handles the sensing internally. This approach also allows for the use of a fixed device which senses the motion of a user based on laser speckle; similar as interaction sensing enabled through a fix installed depth camera (e.g., [15]). With the XWand, Wilson and Shafer introduce a pointing device with the shape of a short

stick, which allows users to point to and interact with objects in a smart environment [33]. This system determines with which object a user intends to interact based on data provided by different integrated sensors such as accelerometers. Hence, no information display is required and user feedback is provided implicitly.

2.3 Mobile Interaction with Remote Screens

For interacting with remote screens, mobile phones have been used in many different ways. For instance, different works examined the potential of gestures where the user pretends to throw an object towards the remote screen while holding the mobile phone in their hand (e.g., [8, 10]). Yet this only allows for imprecise interactions, and selecting items is not possible. Also, different television set manufacturers provide applications for mobile phones which replace the standard remote control: the mobile phone provides an interface for selecting the channel and controlling the volume (e.g., [12]). However, interaction is limited to these simple actions and to steering a cursor through the phone's touch screen. Boring et al. presented a unidirectional mobile phone camera-based approach for selecting and transferring data from a remote display to a personal mobile phone [4]. Different works investigated the potential to control a virtual cursor on the remote screen, using the phone camera by tracking the optical flow (e.g., [13]), or the phone's accelerometers (c.f., [6]). However, this approach is limited in terms of interaction speed. Boring et al. presented TouchProjector, which allows users to perform touch input on their mobile devices; the touch events are then transmitted to a remote screen [5]. Baur et al. investigated interaction based on the metaphor of virtual projection whereby the phone camera captures an external screen to sense the spatial relation of the phone to the screen. Users can exchange data and interact with virtual objects on the remote screen within the camera frustum by performing touch interactions on the mobile phone [3].

The PointerPhone approach presented in this paper, in contrast to the aforementioned works, allows to interact directly with specific targets on the remote screen by pointing in a natural way. Users do not have to hold the mobile phone like a camera as the case when using TouchProjector for remote touch interaction. Also, in contrast to camera-based approaches for steering a cursor, which only allow indirect pointing, PointerPhone allows pointing that is natural for the user, fast, and direct.

2.4 Mobile Phone as Pointing Device

Rukzio et al. investigated users' preferences of mobile interaction with smart objects in their environment [24]. They compared the interaction styles of touching (using near field communication), scanning (based on Bluetooth), and pointing. For pointing, they attached a laser pointer to a mobile phone for interacting with smart objects such as a CD player. Pan et al. investigated how inertial sensing of a phone's position and rotation in relation to a projected remote screen can be used to support pointing using a mobile phone [20]. In their work, Rashid et al. investigated the differences of proximal and distal selection of targets using a mobile phone connected to a Wii controller

[22]. They found that distal interaction is preferred for the selection of targets and results in lower error rates compared to proximal selection. While this work provides interesting insights regarding the selection of targets, they did not focus on other aspects such as interaction with widgets or data sharing interactions.

These works investigated the use of a mobile phone combined with different pointing capabilities. However, they all do not consider the phones' specific attributes and capabilities that allow for novel interactions. PointerPhone considers the characteristics of mobile phones in terms of available sensing, input and output options, and application possibilities which have not been investigated so far.

2.5 Collaboration Support

Work on single-display groupware [30] investigated how multiple users can simultaneously interact with a single display, for instance, by using multiple mouse cursors [21]. However, in such a setting user collaboration is limited to simultaneously pointing to targets on a shared screen. Sharing personal data is not possible, although feasible through the PointerPhone approach. Connecting personal and mobile devices to a remote shared screen to support collaboration has been investigated in different works. The combination of such different devices enables multiple users to access a shared device while each also controlling their personal device [23]. Myers et al. connected multiple PDAs to a central workstation [17], allowing for data exchange. Additionally, the combination of personal and mobile devices with horizontal tabletop-like displays has been studied [26, 28]. Vogt et al. compared mouse-based and laser pointer-based interaction regarding their ability to support collaboration in small groups [32].

3 Phone Pointer Interactions

The underlying concept of using the mobile phone as a pointing device for direct interaction with remote displays is simple but at the same time versatile: users point towards targets on a remote screen in order to perform an action that is applied as the user triggers the action (e.g., selecting or editing an item, controlling widgets). The available hardware of the mobile phone and the remote display yield a number of basic attributes and possibilities for interaction. Basic attributes include whether the user is pointing and the location on the remote display where the user is pointing to. Further, each mobile device that is used as a pointing device can be distinguished through its ID. Accordingly, different users can be distinguished.

Each of these basic attributes can be used for interactions. Considering attributes isolated and combinations of different attributes result in interactions of diverse complexity. Hence, simple interactions can be used as building blocks for more complex interactions. For instance, selecting a target by *pointing* and *clicking* can be part of any activity such as selecting a file on the remote screen. Again, this activity can be part of many applications such as photo sharing or web browsing.

Accordingly, the diverse interaction possibilities can be classified into three layers of abstraction: low-level input and output, widget-level interactions, and high-level interaction and applications. This classification was chosen as it shows which options for interaction are available and how these can be integrated into application designs.

3.1 Low-Level Input and Output

Low-level Input Options. The most basic options for performing input on the mobile phone while pointing to a target on the remote display are using software buttons displayed on the phone's screen (see Figure 2(a)), using hardware buttons available on the phone's case (e.g., buttons commonly provided to control audio volume) (see Figure 2(b)), and performing gestures on the phone's touch screen (see Figure 2(c)). These options can be applied in flexible ways as they can be used either with one hand or with two hands.

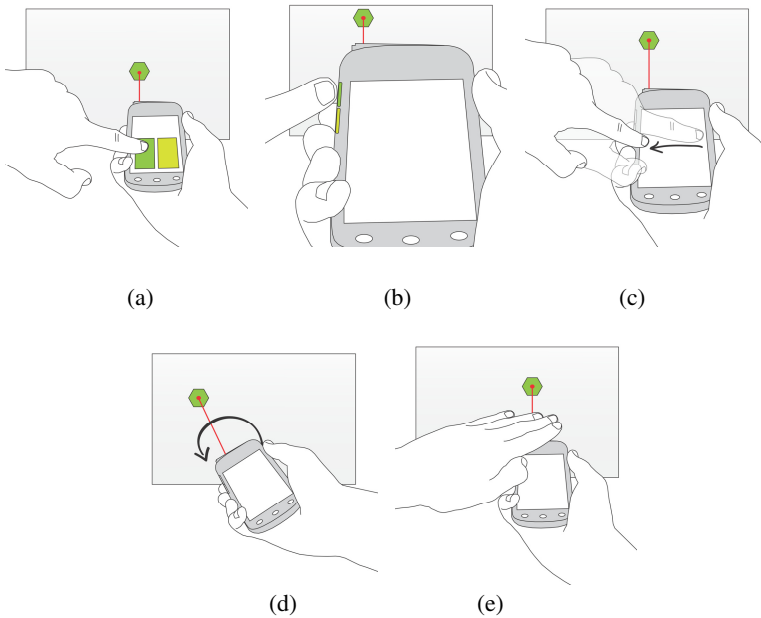


Fig. 2. Basic selection and input options supported through PointerPhone: (a) software buttons; (b) hardware buttons; (c) touch-based gestures; (d) rotation-based interaction; and (e) proximity activation.

One alternative option that avoids pressing hardware or software buttons is to trigger an action by rotating the phone along the pointing axis (see Figure 2(d)). Rotation in different directions (i.e., clockwise and counterclockwise) allows to encode different actions. For instance, a left and right click can be performed. However, rotating the phone could also result in moving the cursor away from the target.

As emphasized by Myers et al. (see [16]), using physical buttons causes unintended jitter effects which could lead to input actions on unintended targets. This is potentially also the case when interacting with software buttons or performing gestures on the mobile phone. One approach that enables users to trigger an action without touching and moving the mobile phone uses the proximity sensor in the mobile phone. While users point to a target, they move their hand close to the phone and trigger the action as their hand gets close enough (see Figure 2(e)). Similarly, users could trigger an action without moving or potentially even without touching the mobile phone through snapping with their available hand which could be sensed using the phone's microphone.

Low-level Output Options. Output options for feedback and information presentation are distributed on the users' mobile pointing device and the remote screen. The latter provides visual feedback and optionally, audio output can be provided (e.g., a television set in the user's living room remotely operated through pointing). The output options of the remote display can be targeted to one specific user only to a limited degree. If several users are using the system simultaneously, for instance, audio feedback provided by the remote screen is audible to all present persons. Visual feedback, however, can be displayed on the remote screen close to a user's pointing cursor in order to make clear at whom the feedback is targeted.

In addition, the personal mobile pointing device enables personal feedback which is not accessible to others. That is, the personal mobile devices provide visual feedback and output on their display, which is visible only to the specific user. Also, audio feedback can be provided either via speakers or headphones, which allow feedback that is not audible to others. Third, mobile devices allow for haptic feedback through vibration.

3.2 Widget-Level Interaction

Often, interaction with diverse applications requires users to specify specific pieces of information or data (e.g., numeric values, strings) in order to control the state of an application. For instance, users control the zoom level of text or specify the volume of audio data using a slider, or select an option from a list using radio buttons. To facilitate this task, many different widgets are available, including as sliders, radio buttons, and text fields, each of which supports the input of a specific data type.

When using mobile phones as pointing device, to interact with applications on a remote or shared display, users require support to interact with all kinds of standard user interface widgets. The given configuration yields up to three different options for interacting with widgets: 1) *rotation* of the mobile phone (see Figure 3(a)), 2) *proximal* interaction on the touch screen (see Figure 3(b)), and 3) *distal* interaction through direct pointing to the widget on the remote screen, selecting it and moving the pointing cursor to change the value of the widget (see Figure 3(c)).

As analyzed by Rashid et al. (see [22]), the performance of proximal and distal selection of targets (e.g., clicking a button) depends on the complexity of the tasks. For complex tasks which involve many and small targets, proximal interaction is superior to distal interaction which was superior in simple tasks (i.e., interaction with few large targets).

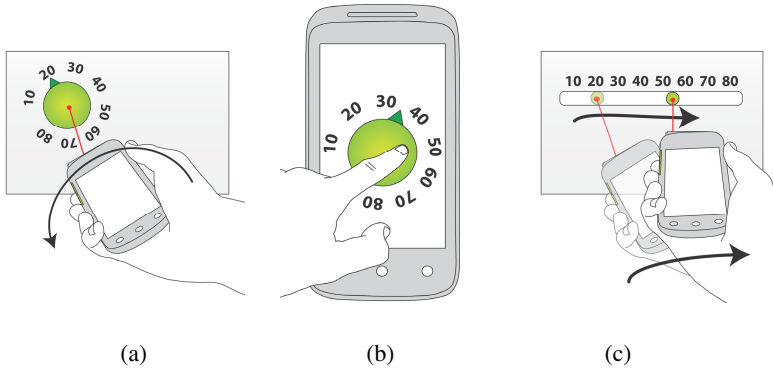


Fig. 3. Controlling widgets for entering data can be implemented in three ways: (a) Rotating the phone to change the value; (b) manipulating the proximal widget representation on the phone screen; (c) distal interaction with widgets on the remote screen.

However, not all three options for interacting with widgets apply to each widget, depending on the type of data supported. Table 1 offers an overview of standard widgets and how they can be controlled through pointing with a mobile phone.

Table 1. Overview of the widget control options

Widget	Orientation	Proximal	Distal
Turning Knob	Yes	Yes	Click & Drag
Sliders	Yes	Yes	Click & Drag
Button	Yes	Yes	Click
Radio Buttons	Yes	Yes	Click
Check Boxes	No	Yes	Click
Text Field	No	Yes	No

Accordingly, only widgets that are designed for the input of continuous values, that is, sliders or turning knobs, could be directly controlled via rotating the phone while pointing to them in order to change their value. Yet it is also possible to control these widgets proximally on the phone touch screen or distally on the remote screen. Standard buttons can be controlled using all three options, given that the phone rotation is

mapped to a selection. Radio buttons could be selected and rotating the phone could change the selection. Check boxes are less suited for this alternative due to their size. Text fields require the user to interact with a keyboard which is most convenient for the user on the mobile phone.

3.3 High-Level Interactions and Applications

In this section, we discuss phone-based pointing interaction techniques that build on the previously discussed interactions.

Data Exchange. Users can share data that is stored on the mobile phone by transferring it to the remote screen. To do so, users select one or several items to share, point to a position on the screen, and trigger the transfer. For instance, users could select an image, point to the desired location, and perform a swipe gesture on the phone towards the remote screen (see Figure 4(a)). On the remote screen the image appears at the location of the pointing cursor (see Figure 4(b)).

In order to receive data from the remote display, users point at the intended item (see Figure 4(c)) and trigger the transfer. As illustrated in Figure 4(d), a swipe gesture on the user's phone could be used to *pull* the item. However, any other low-level input can be applied here.

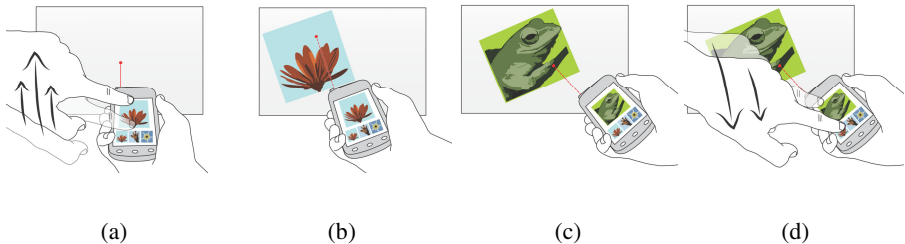


Fig. 4. Transmitting an item from the mobile phone to the remote screen ((a) and (b)) and vice versa picking up an item from the screen ((c) and (d))

Proximal Context Menus. Pointing-based interaction with a remote display through a mobile phone supports the handling of meta information of items such as files which are displayed on the remote display. For instance, context menus are often used in order to change the name of a file. These provide a list of possible options that can be applied to the selected file. Using the mobile phone as pointing device, users first select a file (see Figure 5(a)). The corresponding context menu is then displayed on the mobile phone (see Figure 5(b)), thus for instance facilitating the input of a new file name (see Figure 5(c)). Users are not required to keep pointing at the selected file while using the context menu.

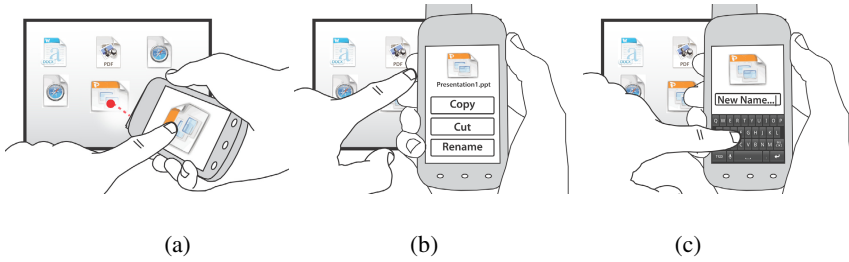


Fig. 5. Proximal context menus: (a) A user selects a file. (b) The context menu is displayed on the phone. (c) A new file name can be typed in.

Drawing and Sketching. Through pointing to the remote screen, users can edit and create graphical content such as sketches. Depending on a selected tool and the corresponding parameter settings (e.g., a brush and a selected color) users can create sketches simultaneously (see Figure 6). As different phones are distinguished, each user can select different tools and settings at the same time.

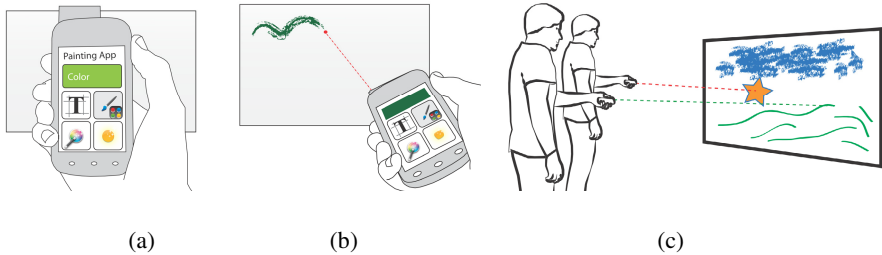


Fig. 6. A sketching application. (a) The phone provides a palette of different tools. (b) Tools are applied through pointing. (c) Multiple users can work simultaneously.

Personal Input and Output. Personal mobile phones as pointing devices allow users to receive personal output (e.g., visual or auditory) as well as to perform input on the personal device in collaborative settings. For instance, when multiple users share a view on a web page, a single user who is interested in following a specific link can point to it and open the corresponding web page on their personal device (see Figure 7(a) and (b)). This allows users to look up additional information without interrupting or disturbing the group activity.

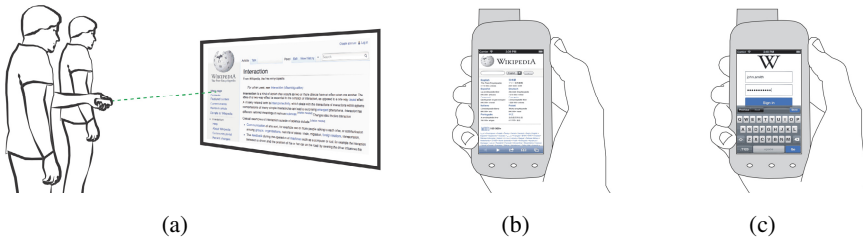


Fig. 7. Input and output on the personal device: (a) To avoid disturbing a group activity, the user may point to a link and (b) open it on their device. (c) Entering information on the personal device.

Additionally, input can be performed on the personal device which, on the one hand, avoids cluttering the remote screen with a large virtual keyboard. On the other hand, input on the personal device allows users to enter sensitive information such as a password. For instance, when a user needs to login to a user account to access some information, the password can be entered on the mobile phone (see Figure 7(c)).

In addition, different types of data such as files, geographical coordinates, contact cards, or appointments can be distinguished and, once selected through pointing to their representation on the remote screen, they can be handled with different applications on the mobile phone. For instance, a user could point to an address and select it, which opens the map application on the mobile phone and displays the given location.

Remote Control. As more television sets support additional diverse applications such as web browsers, one emerging idea is to use secondary display devices to achieve remote control [7, 9, 22]. Using the personal mobile phone for this kind of interaction allows multiple users to interact simultaneously, for instance with web pages displayed on a smart TV (see Figure 8).



Fig. 8. Using the mobile phone as a remote control for browsing web pages on a TV set

3.4 Collaborative Meeting Scenario

To show how different interaction options can be integrated and used as building blocks for a realistic application context, we designed and implemented a collaborative presentation system that supports users in a meeting scenario.

Scenario: Bob is giving a presentation for some colleagues on a large projected remote screen (see Figure 9(a)). Each meeting participant is equipped with a mobile phone which can be used to point to the remote display and control a cursor through phone pointing interaction techniques. Each participant's pointing cursor on the remote screen is distinguished by a different color. Hence, each meeting participant has a visual representation of who and how many users are currently pointing to specific pieces of information on the remote screen. Each projected presentation slide contains

diverse pieces of information. For instance, an overview plan may allow specific views on details of the plan through the selection of a corresponding icon (see Figure 9(b)). This allows users to individually explore and access additional information without disturbing others, as pointing to an icon on the remote screen and selecting it results in a detailed preview on their personal mobile phone. Icons next to a person's name indicate that the contact card can be downloaded to the mobile phone by pointing to it and selecting it. Additional background information can easily be accessed, for example, by pointing to an image on the remote display (see Figure 9(c)). In their meeting, the participants also discuss with each other about the presented topic. This discussion and brainstorming is supported through collaborative sketching on a drawing canvas on the remote screen where to each user can contribute using their phones (see Figure 9(d)).

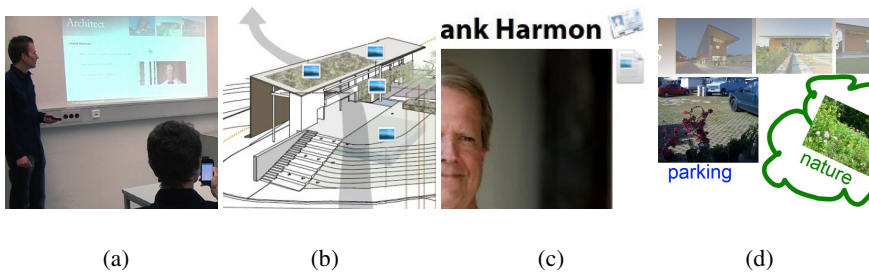


Fig. 9. Collaborative presentation and meeting support (a). Additional information can be accessed through specific icons ((b) and (c)). Users can share data with others on the remote screen (d).

A video illustrates this scenario in detail: <http://youtu.be/qp3pIk1YLxo>

4 Evaluation

In order to gain an understanding on how users would use the system and how they appreciate the different interaction techniques, we performed a qualitative evaluation. The aim was to gain qualitative insights regarding direct pointing-based interaction with a mobile phone and remote display.

4.1 Design

The study session consisted of two parts: After participants were introduced to the study, they performed a series of practical tasks. Tasks that involved collaboration were performed by the participant together with the investigator. After the practical tasks, they filled out a questionnaire. During the session, participants were encouraged to think aloud and continuously talk about their actions. Further, the investigator took notes and the task performance was recorded on video.

Practical tasks. Practical tasks were selected to expose participants to a broad variety of different application contexts. Participants used the PointerPhone prototype for the tasks. The following list of tasks was performed by participants in randomized order.

1. *Browsing.* Participants had to browse through a website that was displayed on the remote screen. The PointerPhone prototype was used for controlling widgets and link selection. They followed text-based instructions, which involved selecting links, downloading images to the phone, and interacting with widgets such as radio buttons through pointing.
2. *Photo sharing.* Selecting and transferring two photos from the phone's library to the remote display and retrieving photos from the remote screen.
3. *Sketching.* Collaborative sketching of a simple building on a shared sketching canvas on the remote screen which required different brushes (selected and configured on the mobile phone). Users also performed text entry on the phone and placing the created text on the sketching canvas for labeling the sketched items.
4. *Completing a form.* Filling in a form on the remote display which included interacting with different kinds of widgets for data input via the mobile phone.
5. *Context Menu.* Renaming, copying, and deleting files displayed on the remote screen by using a proximal context menu on the mobile phone.
6. *Playing.* Playing a simple Pong-like game involving two users who would steer the position and angle of a racket by pointing to the screen to control the translation and rotating the phone to control the angle.

Apparatus. The main components of the apparatus system are mobile phones as pointing devices and a remote display that is connected to a server computer (Figure 10(a)). The mobile devices are connected to the server through Wi-Fi for the exchange of data and commands.

In order to achieve high-level pointing accuracy and low latency, we followed the approach of using laser pointers and camera-based tracking for the pointing task (as previously demonstrated by [16, 19]). That is, a camera is used to capture the remote display. If a user points to the remote screen, the laser pointer creates a bright point on the image which can be extracted through simple image processing. The location of the laser pointer is used to control the user's pointing cursor that is displayed on the remote screen. For distinguishing different laser pointers (and thus different users), a color filter is applied during the image processing step. This tracking approach requires only one calibration sequence before using the system until the camera or display setup is changed (i.e., if they are moved).

We extended a standard mobile phone (a Samsung Nexus S, running Android 2.3) with a laser-pointing module which can be controlled via the mobile phone's software stack (see Figure 10(b)). The laser pointer is turned on and off via a simple circuit with a photodiode that is placed right in front of the flash light LED of the mobile phone. On the Android platform [11], this component can be controlled via a standard application programming interface.

Based on the system prototype, we developed a number of mobile applications for the mobile phone and a corresponding application for the remote screen which implemented all functionalities that were required for the tasks. To allow participants to experience several possibilities of the PointerPhone interaction, the applications provide different options to perform any single action. For instance, users can make selections distally on the remote display, as well as proximally on the phone display. For the sake of consistency, the activation of the pointing (turning on the laser pointer) is the same in all applications. Short tapping on the hardware button on the bottom right activates permanent pointing. Holding the button activates the pointing until its release.

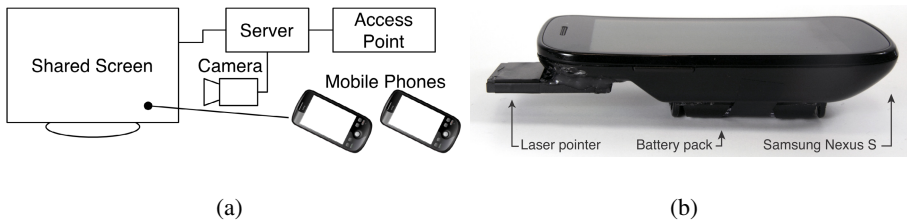


Fig. 10. (a): System components schema. (b): Mobile phone prototype.

Participants. We recruited 14 participants (7 female) aged between 20 and 31 (Mean=26). Of these, 10 were students (diverse backgrounds) and 4 were employees. After the study, they were rewarded for their effort with 10 Euro.

4.2 Observations and Design Implications

In the following, we discuss the results of the feedback sessions and, where possible, summarize findings as a set of design implications that support application designers when considering pointing interactions for their work.

1. *One- and two-handed interaction.* We observed that, during their interaction with the different applications, participants switched between using one or two hands to hold the mobile phone depending on the task. For instance, during the sketching task, 7 of the 14 participants held the phone with two hands. An additional 4 participants held the phone in the right hand and supported it with the left. Only 3 participants used one hand to hold the phone and point during the sketching task. In contrast, during the data-sharing task, 13 participants used a single hand to hold the phone, of whom 5 used the thumb of the same hand and 8 used the index finger of their other hand to interact with the phone interface. Accordingly, the manner in which users choose to hold the phone depends on the given task. For tasks that require precise pointing input (e.g., sketching), users tend to use two hands. Tasks that require less precise pointing, however, lead users to prefer one-handed operation. Hence, interfaces for the hand-held pointing device should encourage two-handed interaction when designed for tasks that require precise pointing. Inversely, interfaces for simple tasks should be adapted to one-handed usage.

2. *Selecting targets.* During the browsing task, participants could select targets such as links either distally on the remote screen or proximally on the phone's display. We observed that participants preferred to select large targets distally while small targets (e.g., text links) led to a preference to select proximally. This concurs with the findings of Rashid et al. who investigated distal and proximal target selection [22]. We observed that distal selection forced users to switch their focus from the remote display to the phone to ensure hitting the correct button. Accordingly, user interfaces for proximal interaction should be designed to allow users to keep their focus on the remote display – for instance, through the use of hardware buttons on the phone if available or a single large software button, so that the user does not have to look at the pointing device.
3. *Navigating.* When users selected an area of the remote screen that should be displayed proximal on the phone, several participants tried to interact with the proximal representation like they were used to interact with smartphone web browsers that allow navigation through dragging and zooming in and out. However, this applies only to small adjustments. Participants expressed that they can select easily an area through pointing to it which is more comfortable than navigating on the phone screen.
4. *Providing output and feedback.* We observed several times that users would focus on the remote display while they select a target there, yet they were not aware of the resulting change on the mobile phone. Inversely, this phenomenon was also observed when a user focused on the phone and performed an action which resulted in an event on the remote display. Hence, it is essential to provide cues (e.g., audio feedback or vibration) which notify users regarding resulting actions. Several participants raised the general point that the remote display should not be used to display user-specific information that is not intended for all users. For instance, when interacting with a web page, users could display tool tip information through pointing at an item for a few seconds. These should be displayed on the personal device.
5. *Controlling pointing actions.* Participants had to manually enable or disable the pointing mode through toggling a button (i.e., turning on and off the attached laser pointer). However, when a participant was engaged with performing a task, they forgot to turn the pointer on which resulted in confusion. Hence, if application workflows allow to anticipate when pointing is required, the system should automatically do so. For instance, when pointing to a web page on the remote display in order to transfer a clipping to the phone for further inspection, the pointing should be disabled automatically to prevent unintended updates.

All users indicated that they liked that they could see a cursor where they were pointing at. This indicated that alternative implementations (e.g., inertial sensing [20]) should provide a visual cursor throughout the interaction.

5 Discussion and Conclusion

In this work, we analyzed options for interaction when using a mobile phone as a pointing device for interaction with a remote display. Related work on pointing interaction, the use of mobile phones for controlling content on public and shared remote displays, as well as collaboration provides a large body of research on interaction techniques for specific contexts. Using the mobile phone as a personal pointing device provides not only powerful computing and sensing technologies but also the user's personal data context such as photos, calendars, and messages. These rich options for interaction are likely to attract the design of applications in the future. Hence, application designers considering PointerPhone applications should be supported through a design space and corresponding guidelines for using aspects from the design space.

Different options exist regarding possible implementations for the application of phone pointing-based interaction outside the laboratory setting. Using inertial sensing to determine the phone's pointing direction is a promising approach, as most available smart phones are equipped with the required sensors (i.e., accelerometer and gyroscope). However, each time a user intends to use such a system, the user needs to calibrate their phone to determine the pointing direction. Moreover, during the interaction the calibration may have to be repeated to maintain the pointing accuracy. Alternatively, direct pointing using a laser pointer can be easily added to standard smart phones. For instance, laser pointers can be plugged into the audio jack of the phone [29] and operated through an application. This approach would require the remote screen to be equipped with camera tracking. As an alternative, the laser pointer could be based on infrared light which can be sensed by specific screens (e.g., Microsoft PixelSense [14])

The investigation of the design space provides the list of basic attributes and characteristics. Further, we provide a classification of low-level interaction options regarding input and output options, widget-level interaction techniques, and high-level interaction techniques and applications. This classification into three levels of abstraction is not fixed and can be extended as it includes only selected examples for applications, and technological features for input are likely to be extended. These techniques can be used as building blocks for complex applications.

Finally, we showed how several interaction techniques can be integrated into a presentation application for a collaborative meeting context. We used a prototype implementation of the system based on mobile phones combined with laser pointers to realize a number of applications that we used for a qualitative study. Results from the study support a collection of five design recommendations that should be considered for the design of pointing-based applications.

Acknowledgements. This work has been conducted in the Emmy Noether research group Mobile Interaction with Pervasive User Interfaces funded by the German Research Foundation.

References

1. Alt, F., Kubitzka, T., Bial, D., Zaidan, F., Ortel, M., Zurmaar, B., Lewen, T., Shirazi, A.S., Schmidt, A.: Digifieds: insights into deploying digital public notice areas in the wild. In: MUM 2011, pp. 165–174. ACM, New York (2011)
2. Ballagas, R., Borchers, J., Rohs, M., Sheridan, J.: The smart phone: a ubiquitous input device. *IEEE Pervasive Computing* 5(1), 70–77 (2006)
3. Baur, D., Boring, S., Feiner, S.: Virtual projection: Exploring optical projection as a metaphor for multi-device interaction. In: CHI 2012. ACM, Austin (2012)
4. Boring, S., Altendorfer, M., Broll, G., Hilliges, O., Butz, A.: Shoot & Copy: Phonecam-based information transfer from public displays onto mobile phones. In: *Mobility 2007*, pp. 24–31. ACM, New York (2007)
5. Boring, S., Baur, D., Butz, A., Gustafson, S., Baudisch, P.: Touch projector: mobile interaction through video. In: CHI 2010, pp. 2287–2296. ACM, New York (2010)
6. Boring, S., Jurmu, M., Butz, A.: Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In: *OZCHI 2009*, pp. 161–168. ACM, New York (2009)
7. Cesar, P., Bulterman, D.C., Jansen, A.J.: Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content. In: Tscheligi, M., Obrist, M., Lugmayr, A. (eds.) *EuroITV 2008*. LNCS, vol. 5066, pp. 168–177. Springer, Heidelberg (2008)
8. Chehimi, F., Rukzio, E.: Throw your photos: an intuitive approach for sharing between mobile phones and interactive tables. In: *UbiComp 2010 Adjunct*, pp. 443–444. ACM, New York (2010)
9. Cruickshank, L., Tsekleves, E., Whitham, R., Hill, A., Kondo, K.: Making interactive tv easier to use: Interface design for a second screen approach. *The Design Journal* 10(3), 41–53 (2007)
10. Dachsel, R., Buchholz, R.: Natural throw and tilt interaction between mobile phones and distant displays. In: *CHI EA 2009*, pp. 3253–3258. ACM, New York (2009)
11. Google Inc. *Android developers* (2013), <http://developer.android.com/index.html> (last access January 15, 2013)
12. Google Inc. *LG TV remote 2011 - Android Apps on Google Play* (2013), <https://play.google.com/store/apps/details?id=com.clipcomm.WiFiRemocon&hl=en> (last access January 15, 2013)
13. Jeon, S., Hwang, J., Kim, G.J., Billinghamurst, M.: Interaction with large ubiquitous displays using camera-equipped mobile phones. *Personal Ubiquitous Comput.* 14(2), 83–94 (2010)
14. Microsoft. *Welcome to Microsoft Pixelsense* (2013), <http://www.microsoft.com/en-us/pixelsense/default.aspx> (last access January 15, 2013)
15. Morris, M.R.: Web on the wall: insights from a multimodal interaction elicitation study. In: *ITS 2012*, pp. 95–104. ACM, New York (2012)
16. Myers, B.A., Bhatnagar, R., Nichols, J., Peck, C.H., Kong, D., Miller, R., Long, A.C.: Interacting at a distance: measuring the performance of laser pointers and other devices. In: *CHI 2002*, pp. 33–40. ACM, New York (2002)
17. Myers, B.A., Stiel, H., Gargiulo, R.: Collaboration using multiple PDAs connected to a pc. In: *CSCW 1998*, pp. 285–294. ACM, New York (1998)
18. Nintendo. *Wii u official site at Nintendo* (2012), <http://www.nintendo.com/wiiu> (last access December 18, 2012)
19. Olsen Jr., D.R., Nielsen, T.: Laser pointer interaction. In: *CHI 2001*, pp. 17–22. ACM, New York (2001)

20. Pan, G., Ren, H., Hua, W., Zheng, Q., Li, S.: Easypointer: what you pointing at is what you get. In: CHI EA 2011, pp. 499–502. ACM, New York (2011)
21. Pawar, U.S., Pal, J., Gupta, R., Toyama, K.: Multiple mice for retention tasks in disadvantaged schools. In: CHI 2007, pp. 1581–1590. ACM, New York (2007)
22. Rashid, U., Kauko, J., Häkkinä, J., Quigley, A.: Proximal and distal selection of widgets: designing distributed ui for mobile interaction with large display. In: MobileHCI 2011, pp. 495–498. ACM, New York (2011)
23. Rekimoto, J., Saitoh, M.: Augmented surfaces: a spatially continuous work space for hybrid computing environments. In: CHI 1999, pp. 378–385. ACM, New York (1999)
24. Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., Chin, J.: An experimental comparison of physical mobile interaction techniques: touching, pointing and scanning. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 87–104. Springer, Heidelberg (2006)
25. Schmidt, D., Molyneaux, D., Cao, X.: Picontrol: using a handheld projector for direct control of physical devices through visible light. In: UIST 2012, pp. 379–388. ACM, New York (2012)
26. Schmidt, D., Seifert, J., Rukzio, E., Gellersen, H.: A cross-device interaction style for mobiles and surfaces. In: DIS 2012, pp. 318–327. ACM, New York (2012)
27. Shizuki, B., Hisamatsu, T., Takahashi, S., Tanaka, J.: Laser pointer interaction techniques using peripheral areas of screens. In: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2006, pp. 95–98. ACM, New York (2006)
28. Sugimoto, M., Hosoi, K., Hashizume, H.: Caretta: a system for supporting face- to-face collaboration by integrating personal and shared spaces. In: CHI 2004, pp. 41–48. ACM, New York (2004)
29. Tangram. Tangram - Smart Dot Pro (2013), <http://www.tangramdesignlab.com/smartdot/en/> (last access January 15, 2013)
30. Tse, E., Histon, J., Scott, S.D., Greenberg, S.: Avoiding interference: how people use spatial separation and partitioning in SDG workspaces. In: CSCW 2004, pp. 252–261. ACM, New York (2004)
31. Vogt, F., Wong, J., Fels, S.S., Cavens, D.: Tracking multiple laser pointers for large screen interaction. In: Extended Abstracts of ACM UIST 2003, pp. 95–96 (2003)
32. Vogt, F., Wong, J., Po, B.A., Argue, R., Fels, S.S., Booth, K.S.: Exploring collaboration with group pointer interaction. In: CGI 2004, pp. 636–639 (June 2004)
33. Wilson, A.D., Shafer, S.: XWand: UI for intelligent spaces. In: CHI 2003, pp. 545–552 (2003)
34. Zizka, J., Olwal, A., Raskar, R.: Specklesense: fast, precise, low-cost and compact motion sensing using laser speckle. In: UIST 2011, pp. 489–498. ACM, New York (2011)