Throw and Tilt – Seamless Interaction across Devices Using Mobile Phone Gestures

Raimund Dachselt, Robert Buchholz

Department of Simulation and Graphics, Faculty of Computer Science
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
dachselt@acm.org, rbuchhol@cs.uni-magdeburg.de

Abstract: To overcome limitations of small screens and to provide intuitive ways of interacting with personal data, this work addresses the seamless combination of sensor-enabled phones with large displays. An intuitive basic set of tilt gestures is introduced for a stepwise or continuous interaction with both mobile applications and distant user interfaces by utilizing the handheld as a remote control. In addition, we introduce throwing gestures to transfer media documents to a large display. By means of these gestures, we also propose transferring a running interface from a mobile phone to a large screen (to improve usability) and back (to achieve mobility). We demonstrate the feasibility of the interaction methods with several application prototypes facilitating a very natural flow of interaction.

1 Introduction

The availability of mobile phones and variety of mobile applications has increased dramatically over the last decade. Nowadays, most mobile phones are able to store and manage all kinds of personal information and media. Despite the advances in the areas of flexibility and usability of mobile phones, all of these devices are inherently limited in terms of their screen size and amount of information they can display.

When combined, two separate technological advances could overcome this limitation. The first one is the increasing integration of powerful sensor packs into mobile phones. The potential of creating interaction techniques involving these sensors has been recognized in research for some time, but is still largely uncommon. The second advance is the increasing usage of large-screen displays for home entertainment and in public spaces. Nowadays, in both settings, the screens are still driven by conventional computers using traditional input devices or TV-style remote controls.

The goal of this work is to combine the expressiveness of interaction enabled by sensor equipped mobile phones with large displays in a way to seamlessly couple screens of various sizes. One approach is to use intuitive throwing gestures to transfer personal media data such as music, pictures, diary entries and map locations to a large screen. An even more advanced approach is to seamlessly transfer the whole user interface between various devices. With this approach, the mobile phone is no longer restricted to a small computer with a very limited display. It can extend its role to that of an input and control device for a distant large screen by using a simple, intuitive, and generic set of tilt gestures to ensure natural interaction without a steep learning curve.

The remaining paper is structured as follows. After presenting related work, section three introduces the basic concept of the *tilt and throw interaction*. Section four illustrates these techniques with application scenarios, followed by some notes on their implementation. The paper is concluded with a summary and outlook to future work.

2 Related Work

One of the first researchers to describe the usage of tilt as an input method for small screen devices was Rekimoto [Rek96]. Early devices had no built-in accelerometers or other sensors. Therefore, researchers manually added sensor packs to palmtop computers and mobile phones, e.g. for tilt-based browsing of photographs [CC+07]. Cho et al. conclude that browsing with a single tilt gesture is indeed feasible and appealing to users. Others used the built-in camera of mobile phones as an alternative to facilitate movement or position detection, e.g. [WZC06, BRS05]. A typical application of using accelerometer sensors is the training and recognition of gestures on mobile phones, e.g. [KK+06], as well as on the Nintendo Wiimote. In contrast, we focus on very few simple gestures that do not require machine learning for training and require almost no learning curve. Another application of using motion sensor data is fast text input using predefined tilt gestures [WB03]. The improvement of input speed comes with the drawback of having to memorize complex gestures.

A creative way of using PDA-attached sensor packs is to not only detect a throwing gesture, but to also measure its force and direction in order to determine which device was targeted [YT+05]. This approach simplifies the connection process even further, but it currently requires additional stationary equipment.

Amongst others, Cheverst et al. evaluated the interaction of mobile phones with situated displays [CD+05]. However, their public displays act as whiteboards and are places to retrieve or drop information and not facilities to extend a user interface to. Pering et al. also mentioned the idea of using a mobile phone to control other devices and even transfer media to these [PAW07], but their main focus is on simplifying the process of connecting to these devices, not interacting with them.

3 Basic Concept of Tilt and Throw Interaction

In the domain of personal media management, many tools rely on a simple set of interactions. They can be found in higher-level interaction tasks such as scrolling, zooming, panning, rate control, or menu selection. One example are mobile music players with hierarchical lists of music items which can be navigated using the up and down directions for scrolling and right and left directions for working with sub menus. Another example is the Windows Media Center[1] application, serving as a home entertainment hub for TV, video, DVD, photos and music. It heavily relies on a crossbar menu concept also supporting remote control operation using directional keys.

3.1 Tilt Interaction with mobile and distant interfaces

We identify the following basic set of interactions: *up, down, left,* and *right* movements of either a cursor position, some widget highlight (such as in menus), or a direct mapping to navigate a movable application space (e.g. maps). These four movements can either be *discrete (stepwise)* or *continuous (fluent)*. For the latter we distinguish between a linear and non-linear mapping of the movement to the resulting interface changes.

Figure 1: Tilting along the x axis: *down, neutral, up* and along the y axis: *left, right*.

The idea is to support this simple set of interactions by recognizing corresponding tilt gestures with an off-the-shelf sensor-enhanced mobile phone (compare Figure 1). Such modern devices offer a built-in triaxial accelerometer measuring the forces along the three coordinate axes. These force values can be interpreted as a vector pointing to the center of earth. For many applications it is even sufficient to directly interpret the x and y sensor values as measures for the rotation along the x and y axes.

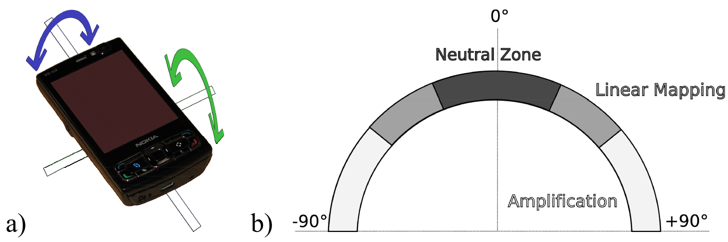

Figure 2: a) Tilting along the x and y axes. b) Different mapping zones used for each axis.

Figure 2 a) depicts the two axes and supported tilting movements. While holding the mobile phone in a resting position on his palm, the user can easily perform tilting gestures in all directions along the x and y axes. For performing *discrete interactions*, the phone is simply moved back to the neutral resting position (see Figure 1) after each stepwise interaction. For this type of interaction a simple threshold is used to recognize a single interaction in either direction. For the *continuous interaction* mode a neutral zone of about +/– 5° should be defined. This compensates for trembling of the user's hand and accidental movements. Compare Figure 2 b) for a schematic depiction of how rotational values of one axis are divided into zones which transform the values differently. Beside

[1] Windows Media Center http://www.microsoft.com/windowsxp/mediacenter/

the neutral zone either one or two additional zones are used. Each zone can either have a linear mapping of the rotational values to the user interface or a non-linear, usually exponential function amplifying the movement. Two zones allow for fine control in the near field and accelerated control for distant parts, e.g. for scrolling documents. A good transfer function should ignore very small sensor values, accurately reflect medium ones and amplify big ones. We designed our implementation to allow for a flexible assignment of the zones and transfer functions depending on the application.

Like previous research suggested (e.g. [Rek96] or [CC+07]), the tilt gestures can be used to control applications on the mobile phone itself, e.g. for browsing photographs. To avoid an unfavorable viewing angle and to overcome the display limitations of small screens we propose to transfer the (same) interface to another computer with a bigger display at some distance and to use the mobile phone as a remote control operating the UI. By providing the described set of very basic gestures, attention does not need to be split, and users can concentrate on the large display only.

3.2 Throwing data and interfaces to large displays

In the domain of personal media management, items frequently need to be transferred between various devices, e.g. camera phones, home servers, or mobile music players. To provide an intuitive way of transferring personal data, we suggest simple throw gestures to connect a mobile device and a large display (i.e. stationary computer) and transfer data. One usage scenario is the transfer of newly taken pictures to a home media center: Performing a *throw gesture* with the phone (see Figure 3, center) within range of the computer transfers all data to the TV screen / HDTV projection over WiFi or Bluetooth. This connection is already established while entering the room. Afterwards, the user can browse or zoom the pictures on the large display by tilting the mobile phone, thereby experiencing a seamless interaction across devices.

In the opposite direction, i.e. from a big screen to the mobile phone, we suggest a *fetch-back-gesture* by rapidly moving the phone down on the z axis. It can be thought of preparing to put the device in your trouser pocket or lay it down on the table for taking it away later. This gesture is asymmetric to the throw gesture for two reasons: one is a missing metaphor for the opposite of throwing; the other is the necessary distinction to be made between recognizing the tilting and fetch back gestures.

To recognize a throwing gesture, we currently use a threshold of the Euclidean length of the raw force vector delivered by the sensor. In tests we determined a threshold of 180 as a compromise between preventing false-positives and the momentum necessary to be recognized as a throwing gesture.

Besides throwing or fetching back your data, we contribute the concept of *transferring a running user interface* from a mobile phone to a large screen (to improve usability) and back (to achieve mobility), again by using throwing gestures. This provides a way of interacting in a truly seamless fashion across devices, assuming that an application exists for both the mobile and stationary device. The state of an application running on the mobile phone needs to the transferred to a large screen. If the basic tilt interactions are supported by the program, the user can immediately continue using the application on the large display. Since the interaction method, i.e. tilting the handheld, stays the same, throwing the interface can be conceptually seen as just switching displays. However, unlike VNC-enabled remote screens the interface might look different on both devices.

4 Application Scenarios

Currently, we have implemented three applications employing the previously described techniques. They serve as a test bed for evaluating and refining the gesture interactions.

Image/Media Browser: As described in the previous section, this application allows throwing images or other captured media experiences from your mobile phone to a large, distant display, e.g. a home cinema projection. On connecting, all necessary data is received and cached by the PC based on a MD5 hash for later use. While using only the phone, the image browser is limited to simple full screen or tiled picture browsing. When transferring control to the large screen, OpenGL cover-flow browsing can be used and additional information (e.g. EXIF data) can be displayed. The cover flow view can be navigated smoothly with phone tilt gestures.

The zoomable music browser MAMBO: This music browser, available on mobile and other devices, supports visual browsing and faceted filtering of songs, albums, or artists according to different hierarchical metadata facets [DF07]. Thus, scrolling long lists of items is entirely eliminated. The stepwise navigation through and within levels of a facet hierarchy is achieved by *discrete directional tilt gestures* (left and right for panning, up and down for zooming in or out). Figure 3 (left) displays the Mambo UI on a handheld device[2] being used with tilt gestures. After throwing the interface to the distant display as shown in the middle, Mambo can be remotely operated as depicted on the right.

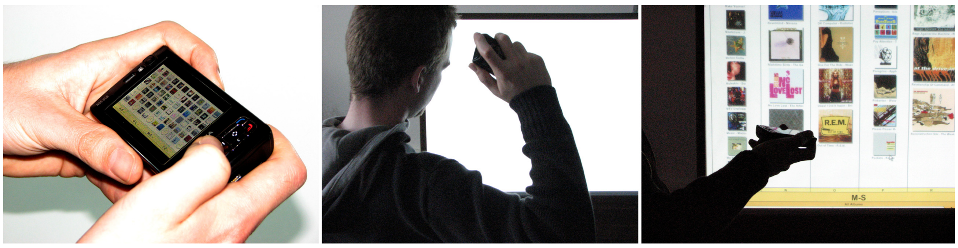

Figure 3: Operating Mambo on a handheld device | throwing it to a distant display | using tilt gestures to operate the interface on a large screen.

Geo-referenced data on maps. The idea in this scenario is to use GPS-enabled location tagging of photos on current mobile phones. When coming back from a trip, a user can throw the geo-referenced captured media to a large screen and subsequently see it displayed at the appropriate locations on a map. We implemented a first prototype using the

[2] Please note that Mambo currently does not run on a Symbian phone, but on other handheld devices. However, using the phone as a remote control to interacting with Mambo on a large screen was already implemented.

Google Earth COM API. With it, one can remotely operate the Google Earth interface and perform the interactions depicted in Figure 4: Panning of the map is simply performed by a *continuous tilt interaction* of the handheld device in the corresponding direction. To facilitate the various interaction modes provided with 3D map programs, mode switches are required. Pressing the central button on the phone, users can switch between *pan mode* and *zoom and tilt mode*. In the latter, zooming in is performed in a continuous way by moving down the phone (see Figure 4, center), zooming out by tilting it in the opposite direction. Changing bird's eye view to street view, i.e. tilting the map, is achieved by tilting the phone to the left or right, which becomes more natural when rotating the phone about 90° to a position as shown in Figure 4 on the right.

Figure 4: Remotely operating Google Earth with a mobile phone | zooming in | tilting the surface.

5 Implementation

For our prototypes we are using a Nokia N95 8 GB Symbian S60 phone. We chose Python as the programming language for the phone, since it allows rapid prototyping. Besides using the Python interpreter and library for Nokia's S60 series, we utilized the N95AccelerometerPlugin provided by the Nokia Research Center as well as the aXYZ sensor extension for Python. The N95 is equipped with a simple triaxial accelerometer. The uncalibrated sensor delivers data which is subject to some jitter. This can be compensated by some means of smoothing (average, moving average, median, ...) on the phone at the cost of reducing the temporal resolution or accuracy. A simple or composite transfer function to map the sensor data to meaningful interaction (e.g. zoom) values is executed on a PC after the sensor data has been transmitted. The N95 supports communication over Bluetooth and WiFi. We decided to use WiFi exclusively, because there is no standard Bluetooth API for Windows. Employing WiFi, we used the low-latency UDP for transmitting sensor data, and the reliable TCP to transmit files, e.g. images.

6 Conclusion and Future Work

In this work we investigated a basic repertoire of intuitive tilt interactions on accelerometer-enabled mobile phones. The continuous and stepwise tilt gestures can be mapped to a variety of elementary interactions often used in media-centered applications, among them zooming, panning, scrolling, or menu selection. The simple but powerful gestures can be used to control various applications both on mobile screens and large displays. In addition, we proposed simple throw gestures to connect a mobile and distant display by transferring data and even running interfaces, thus facilitating a natural flow of interaction. With these gestures, mobile devices and large displays are combined in a seamless manner. We also implemented several prototypes, which need to be refined and carefully evaluated as future work. We will closer investigate the transfer of interfaces, which is a non-trivial software engineering and usability challenge, and examine additional application scenarios combining private mobile and public displays including privacy issues.

Acknowledgements. We wish to thank Jürgen Scheible for the inspiration to parts of this work by his pieces of art including throwing videos with mobile phones (see http://www.leninsgodson.com/artblog/). Our work was funded by the "Stifterverband für die Deutsche Wissenschaft" from funds of the Claussen-Simon-Endowment.

References

[BRS05] Ballagas, R., Rohs, M., and Sheridan, J. G. 2005. Sweep and point and shoot: phonecam-based interactions for large public displays. In CHI '05 Extended Abstracts on Human Factors in Computing Systems (Portland, OR, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, 1200-1203.

[CC+07] Cho, S., Choi, C., Sung, Y., Lee, K., Kim, Y., and Murray-Smith, R. 2007. Dynamics of tilt-based browsing on mobile devices. In CHI '07 Extended Abstracts on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1947-1952.

[CD+05] Cheverst, K. and Dix, A. and Fitton, D. and Kray, C. and Rouncefield, M. and Sheridan, J. and Saslis-Lagoudakis, G. (2005) Exploring Mobile Phone Interaction with Situated Displays. In: PERMID workshop at Pervasive 2005.

[DF07] Dachselt, R., Frisch, M.: Mambo: A Facet-based Zoomable Music Browser, Proc. of the 6th Int. Conf. on Mobile and Ubiquitous Multimedia (MUM 2007), Oulu, Finland, 110-117, ACM, December 2007.

[KK+06] Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. 2006. Accelerometer-based gesture control for a design environment. Personal Ubiquitous Computing. 10, 5 (Jul. 2006), 285-299.

[PAW07] Pering, T., Anokwa, Y., and Want, R. 2007. Gesture connect: facilitating tangible interaction with a flick of the wrist. In Proceedings of the 1st international Conference on Tangible and Embedded interaction (Baton Rouge, Louisiana, February 15 - 17, 2007). TEI '07. ACM, New York, NY, 259-262.

[Rek96] Rekimoto, J. 1996. Tilting operations for small screen interfaces. In Proceedings of the 9th Annual ACM Symposium on User interface Software and Technology (Seattle, Washington, USA, November 06 - 08, 1996). UIST '96. ACM, New York, NY, 167-168.

[YT+05] Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., Hashizume, H. 2005. Toss-It: Intuitive Information Transfer Techniques for Mobile Devices. In Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005), 1881-1884.

[WB03] Wigdor, D. and Balakrishnan, R. 2003. TiltText: using tilt for text input to mobile phones. In Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, November, 2003. UIST '03. ACM, New York, NY, 81-90.

[WZC06] Wang, J., Zhai, S., and Canny, J. 2006. Camera phone based motion sensing: interaction techniques, applications and performance study. In Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM, New York, NY, 101-110.