TimeTilt: Using Sensor-Based Gestures to Travel through Multiple Applications on a Mobile Device

Anne Roudaut[1,2], Mathias Baglioni[1,2], and Eric Lecolinet[1]

[1] Telecom ParisTech – CNRS LTCI UMR 5141, 46 rue Barrault, 75013, Paris, France
[2] Alcatel-Lucent Bell Labs France, Villarceaux, Route de Villejust 91620, Nozay, France
`{anne.roudaut,mathias.baglioni,`
`eric.lecolinet}@telecom-paristech.fr`

Abstract. We present TimeTilt, a sensor-based technique that allows multiple windows switching on mobile devices, and which overcomes the limitations of mobile devices, i.e. their impoverished input bandwidth (often no keyboard, a small tactile screen and the drawbacks of one-handed interaction). TimeTilt, which is based on a lenticular metaphor, aims at both reducing the activation time when switching between views, and supporting a natural mapping between the gestures and the navigation. We draw a brief classification of sensor-based gestures that could be used in mobile conditions, and we present an experiment.

Keywords: Mobile devices, One-handed interaction, Multiple windows, Sensors, Gestures, Lenticular, Undo, Redo.

1 Introduction

More than simple communication systems, mobile devices provide a full set of multimedia services, helping people to manage medias anywhere. Although mobile device functionalities almost reach the ones of a laptop, they are constrained by their small form factor. Most familiar input devices have been removed (e.g. there is no mouse and often no keyboard) and the lack of screen real estate makes it impossible to display a large amount of information. Moreover, mobile devices are often operated with one hand, by interacting with the thumb [11] (e.g. when the user is standing in the metro). This causes occlusion on the screen and makes selection imprecise. The recent integration of motion sensors creates a great potential for designing new techniques: sensors fit very well with the graspable form of mobile devices and do not hamper touchscreen visualization and interaction. Several studies have been done in this field [7, 3], for instance for recognizing 3D shapes, for scrolling a list by tilting or for enriching games. However, in mobile conditions, users may perform involuntary gestures, and certain gestures may seem awkward to perform in public spaces. A compromise must then be found between gestures that are comfortable for users and compatible with their activities, and that do not hamper usual interaction techniques.

To address these issues, we present TimeTilt, a window management technique that relies on sensor-based interactions. This new technique does not interfere with standard interactions. This feature is especially welcome because it makes it possible to activate commands or trigger interaction techniques in the usual way, while changing

windows easily. This makes it for instance possible to drag and drop objects from a window to another one. In fact, changing windows quickly is useful in a variety of situations, such as switching between an overview and a zoomed view, between tabled views, etc. We first present TimeTilt and its properties. The two kinds of gestures used in this technique (smooth and jerk gestures) are described in the last part.

Smooth movements create a lenticular effect

Back Tap

Jerk movements

Fig. 1. TimeTilt helps to "travel in time" to reach previously opened windows. A tap on the back of the device triggers a continuous mode so that the user can navigate through windows by performing smooth gestures. A discrete mode can be activated with jerk movements.

2   TimeTilt Technique

While this is technically possible, mobile devices do not allow the user to switch easily between several windows. Because there is no room to display more than one window, the user must close the current window, find the other application (s)he now wants to interact with by navigating through the interface, and open it. Some applications may have their data stored in memory, but it is not always the case and the user must wait for the application to be loaded. TimeTilt was developed to solve this problem. Unlike desktop windows managers that have a spatial layout (with icons in the desktop bar), it is based on a temporal approach well suited for mobile devices. TimeTilt allows the user to tilt the device to "travel across time" and to navigate through open windows. When the desired window is found, the user just presses the screen to activate the corresponding application.

TimeTilt is based on a lenticular metaphor (Fig. 1). The lenticular effect (used in certain gift postcards) makes it possible for the user to see different images depending on the card orientation. The principle is to merge multiple images into the same sheet with a strips and lenses mechanism. As one's viewing angle changes, one sequentially sees different strips of the image underneath the lenses, showing an animation. We used this effect to design a user-friendly technique that helps users to perform gestures (as advised in embodied user interface [4]). By analyzing the data provided by a G-sensor (i.e. accelerometers), TimeTilt takes into account three kinds of movements:

- A finger tap at the back of the device triggers/stops a continuous mode
- A smooth tilt of the device allows navigating *continuously* through windows (the lenticular metaphor animation provides a fluid transition between windows). When the user taps the screen, the current application is activated
- A jerk movement over the z-axis (forward or backward) performs a *discrete* action, activating the previous or a next opened window

Among these gestures, we distinguish two categories: the smooth ones as tilting the device for scrolling a list or navigate in a map [1, 5, 10, 12] and the jerk ones, which are sharper. The back tap is a jerk gesture because it creates a large acceleration variation, as we will see. TimeTilt leverages the three gestures in combination.

First, the back tap triggers a continuous mode *on demand*, without using standard inputs (as a physical key which may already trigger an action). The user chooses the beginning of the gesture to have always the screen visible. As showed in [9], users can navigate through 8 to 12 views by tilting the device. Moreover the activation tap is performed on the back of device, unlike [8] where users tap on the sides. This design is adapted to one-handed interaction because when interacting with the thumb, other fingers can activate commands from the back, as in recent studies [2].

Secondly, smooth gestures trigger a continuous mode that is well suited to switch between views. A user can easily navigate back-and-forth between, e.g., a map and an email application. They need not perform time-consuming steps that would distract his attention from the task.

Finally, jerk gestures offer a discrete mode that helps to quickly reach the last opened window: when interrupted by a phone call, the user can return to his previous task without searching in an application list (like in the iPhone) or without having to select small screen widgets with a stylus (as in Windows Mobile devices). This makes our technique well adapted to one-handed interaction.

In summary, this sensor-based gesture vocabulary contains three movements over the z-axis to support a natural mapping with the lenticular metaphor, which helps the user to understand and memorize the gestures. In the following, we present a use case of TimeTilt and then we explain how smooth and jerk gestures are recognized.

3   TimeTilt Use Case

Bob is searching on Google Map when he receives from a friend a message that indicates a date location. Bob copies the street address and taps on the back of his device. The messaging window freezes. From that position and according to a threshold, each rotation of the device around the z-axis displays the previous windows opened. Bob tilts the device forward to make the Google Map window appears and taps the screen. Bob pastes the street address in the Google Map research field, and the destination appears. He can then perform the reverse interaction and return to the messaging window to answer back to his friend.

4   Smooth and Jerk Gestures Features

The acceleration variation vector is a relevant value to distinguish smooth and jerk gestures because it does not depend on mobile orientation. Gestures can be recognized

whatever the device position in the hand. Secondly, its values are almost flat during smooth gestures. Therefore, the challenge is to detect and identify jerk gestures. Because sensor hardware events are "continuous" (more precisely each 50ms in our implementation), there is neither start nor end of a gesture. To address this issue, our algorithm observes the acceleration variation and waits for a stable frame of 300ms. This value detects the jerk gestures studied in the experiment to be described below.

We conducted a preliminary study to collect jerk movements performed over the z-axis. The experiment aimed at finding the characteristics that are relevant to design a recognizer. Our 8 participants (2 female) were asked to perform gestures, while seated, with their preferred hand holding the device. There were six gestures to perform six times: 1) a back tap; 2) a back tap while holding the finger on the device before releasing; 3) a small jerk movement forward and 4) backward; 5) a sharper jerk movement forward and 6) backward. The experiment software was developed in C# (.Net Compact Framework) on a HTC Touch HD (Windows Mobile 6.1) with a 480x800px resistive touchscreen.

ANOVAs helps us indentify six relevant features for classifying jerk movement (p<0,001): 1) the global amplitude ($F_{5,35}$=21.5); 2) the maximum amplitude between two consecutive values ($F_{5,35}$=12.2); 3) the total duration ($F_{5,35}$=29.7); 4) the duration between the maximum and the minimum value ($F_{5,35}$=4.3, p<0.003); 5) the number of sign changes ($F_{5,35}$=5.6); 6) the sign of the first acceleration episode ($F_{5,35}$=15.7). Post-hoc multiple means comparison tests showed that back taps are faster, with a strong acceleration that does not seem to occur when performing the other jerk movements. We also observed that the shape of these gestures differed to a sufficient extend to avoid confusion with the natural gestures performed by the user. Finally, further analysis showed that although values differ from a user to another, patterns are similar enough for allowing proper recognition.

In the future we plan to discriminate more gestures, and to combine several sensors (e.g. with microphones as in [6]). We will compare TimeTilt with alternative methods to evaluate its performances and the satisfaction of users. We also want to increase TimeTilt capabilities by applying it for repeated undo/redo actions, or by enabling gestures over the x-axis with a "fan" metaphor to switch between tabs for instance.

References

1. Barlett, J.F.: Rock'n scroll is here to stay. IEEE Comp. Graph. Ap., 40-45 (May/June 2000)
2. Baudisch, P., Chu, G.: Back-of-device Interaction allows creating very small touch devices. In: CHI 2009 (to appear, 2009)
3. Dachselt, R., Buchholz, R.: Throw and Tilt - Seamless Interaction across Devices Using Mobile Phone Gestures. In: GI 2008, pp. 272–278 (2008)
4. Fishkin, K.P., Gujar, A., Harrison, B.L., Moran, T.P., Want, R.: Embodied user interfaces for really direct manipulation. Commun. ACM 43(9), 74–80 (2000)
5. Harisson, B., Fishkin, K., Gujar, A., Mochon, C., Want, R.: Squeeze Me, Hold Me, Tilt Me! An exploration of manipulative user interfaces. In: CHI 1998, pp. 17–24 (1998)
6. Harrison, C., Hudson, S.E.: Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile finger Input Surfaces. In: UIST 2008, pp. 205–208 (2008)

7. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing techniques for mobile interaction. In: UIST 2000 (2000)
8. Linjama, J., Kaaresoja, T.: Novel, minimalist haptic gesture interaction for mobile devices. In: NordiCHI 2004, pp. 457–458 (2004)
9. Rahman, M., Gustafson, S., Irani, P., Subramanian, S.: Tilt Techniques: Investigating the Dexterity of Wrist-based Input. In: CHI 2009 (to appear, 2009)
10. Rekimoto, J.: Tilting operations for small screen interfaces. In: UIST 1996, pp. 167–168 (1996)
11. Roudaut, A., Lecolinet, E., Huot, S.: Taptap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens. In: ACM AVI 2008, pp. 146–153 (2008)
12. Small, D., Ishii, H.: Design of spatially aware graspable displays. In: CHI 1997, pp. 367–368 (1997)