

A Comparison of Motion and Keypad Interaction for Fine Manipulation on Mobile Devices

Lasse Schwarten
TZI, Universität Bremen
Bremen, Germany
lasse@tzi.de

Christoph Grimmer
TZI, Universität Bremen
Bremen, Germany
cg@tzi.de

Benjamin Walther-Franks
TZI, Universität Bremen
Bremen, Germany
bwf@tzi.de

Sebastian Feige
TZI, Universität Bremen
Bremen, Germany
sfeige@tzi.de

ABSTRACT

Ever since the introduction of products such as the Nintendo Wii or the Nokia N95, motion-based interaction has become en vogue in research and industry alike. The employment of this new form of interaction is the subject of extensive research, especially for mobile use. Tilt or gesture interaction allow for quick and intuitive manipulation since the device is already at hand.

So far there is neither a consensus on which metaphors can support input based on device motion, nor what types of applications are benefited by this control. Motion-based approaches exist for coarse tasks such as selecting an entry from a menu, but how does such an input mechanism work for finer input? Further, it is important to scrutinise whether this new way of interacting really is an improvement over tried-and-tested keypad interaction.

In order to address these questions, we present a comparison of tilt-based interaction with the established keypad interface for fine manipulation. For this we developed a new metaphor for tilt-based control we call *MarbleControl*, which can be used for a wide range of applications.

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Design Tools and Techniques—*User Interfaces*

General Terms

Design, Experimentation, Human Factors

Keywords

Mobile HCI, motion-based interaction, tilt-based interfaces, one-handed interaction, precise manipulation

1. INTRODUCTION

The field of Mobile HCI has received increasing attention with the growing capabilities of mobile devices regarding sheer computational power as well as larger displays, increased storage capacity and improved sensors. Especially the idea of using accelerometers to control any kind of device has come into focus since the introduction of the Nintendo Wii and its remote (called Wii Remote or Wiimote for short), and has found its way into mobile handsets such as the Apple iPhone. Research in this area focuses either on gesture recognition [2] or directly measuring a device's absolute orientation (roll, pitch and yaw) for scrolling webpages, panning maps [5] or navigating in menus at best.

But what about tasks that require more fine-grained control? So far, procedures such as media editing are limited to controlled environments where mouse and keyboard allow for detailed input. As our lifestyles grow more mobile, and the production and editing of media are spreading to every household, there is a demand to perform operations such as a quick adjustment of a photograph or a tweak of a video you just shot on the go.

One could argue that the predominant form of touch-screen input for PDAs via a stylus or finger solves such issues. But this setup requires two hands – one for holding the device, the other for pointing. Many mobile scenarios exist that don't give users this freedom, such as when holding on to bars in public transport, holding on to shopping or other carried items.

As opposed to gesture input that requires elaborate gesticulation, controls using a device's inclination angle allow for discrete manipulation in two dimensions. However, metaphors are needed that support affordance in interaction based on device motion. To the knowledge of the authors, no such metaphors exist yet.

In this paper we introduce an interface metaphor that combines continuous control (as used for navigating, scrolling, panning) with the versatility of a pointing device. It gives constant visual feedback to the user and is easy to learn as it uses affordances we know from toys: The marble maze game. In order to find how this form of tilt-based interaction compares to the tried-and-tested keyboard interaction, we evaluate how they perform for tasks that require fine manipulation in a mobile setting. We chose mobile media editing as a representative use case.

© The Author 2008.

Published by the British Computer Society

2. RELATED WORK

Over the last decade, new interaction paradigms for mobile devices have become an extensive field of study, an overview of which would exhaust the scope of this paper. Of the various interaction forms examined in research, a very popular one is motion-based or gestural interaction. The convergence of input device and display into one no longer requires the user to mentally transfer between the two. Mäntyjärvi et al. differentiate between gestures and direct measure & control. They classify movement sensor based user interfaces into three categories [3]: 1. Direct measuring of tilting, rotation or amplitude, 2. discrete gesture commands and 3. continuous gesture commands.

Gesture interaction is receiving a lot of attention in mobile HCI research [2, 4]. While Mäntyjärvi et al. correctly state that the complexity of detecting and processing these input patterns is far higher than “simple” measurement of sensor states and direct mapping to applications [3], they seem to downplay that this can nevertheless provide effective and easy-to-use interfaces.

Tilt-based interfaces use the slant or tilt of a mobile device for control. Since their introduction [8], various efforts have been made to improve tilt-based interaction [7, 1, 5]. The basic concept stays the same: to map the change of a relative or absolute angle of inclination onto a continuous action on the screen, such as scrolling up/down a menu or panning a large image or map. Alternatively, “bursts” of tilting can enable a step-by-step navigation of icons/thumbnails, as the products of the company GestureTek¹ demonstrate.

Though vibrotactile feedback has been studied [7], existing tilt-based interfaces all have in common that they are very poor on visual feedback, which is essential if the user is to efficiently control the tilting. For instance, there is no feedback on the extent of the tilt range (or angle), or how far the device needs to be inclined in order to provoke a certain action. Furthermore, the continuous/discrete navigation along the horizontal and/or vertical axis may suffice for strongly structured screen layouts, such as menus or thumbnail grids, but are no true replacement for a true pointing device such as a stylus or a finger. For this, however, the use of a second hand is required.

3. FORMS OF MOBILE INTERACTION

A large number of interaction modes are feasible for mobile devices, their appropriateness depending on hardware at hand, the circumstances in which a mobile device is used and the controlled application. The following is a list of the most popular forms of mobile interaction.

Keypad control is something like the gold standard since it is by far the most established interface for mobile devices. With the term ‘keypads’ we subsume all types of controls that consist of several hardware buttons, including standard mobile phone number keys as well as the more simplified button controls below the touchscreen of modern PDAs. Keypad control supports one- and two-handed interaction and is quite robust to environmental noise.

Touch sensitive surfaces allow for quite precise interaction and offer flexibility similar to other pointer interfaces like a mouse. The main drawback is that a user either has to use a stylus, which enforces two-handed interaction and simply might be dropped or even lost, or has to restrain himself to very imprecise control because of the size of his finger tip.

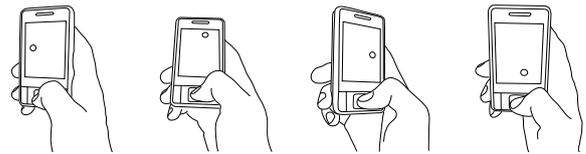


Figure 1: Tilt-based interaction with *MarbleControl*

Trackpoints and Trackballs (as built into the BlackBerry Pearl) offer the flexibility of a pointer interface while still allowing one-handed interaction. Still only very few mobile devices provide the necessary hardware.

Accelerometers in modern handhelds such as the Nokia 95 or Apple iPhone have made *motion-based* interaction possible. This can be further classified into *tilt-based* and *gesture-based* control (see section 2).

Voice control is still very rarely used for interaction with mobile devices, even though it would allow hands-free interaction. But voice recognition requires large computational resources and is very sensitive to background noise.

Next, we present a use case for fine manipulation on mobile devices. With this in mind, we select two forms of interaction of those listed above to be pitted against each other and present how we implemented them for the application at hand.

3.1 Use Case: Mobile Image Editing

Since we want to evaluate precise, one-handed interaction with mobile devices we chose mobile image editing as a use case. Image editing (e.g. cropping a person’s head in a typical mobile phone snap shot) is an example of a combination of coarse and fine manipulation (to adjust the selection marquee’s edges). Therefore we built an application which allowed to load an image, to resize and position a selection marquee, and to crop the selected part. We kept standard operations such as load or exit to be used in conventional means, i.e. by using the device’s soft keys. For moving and scaling of the selection marquee to define the cropped area we developed two modes: moving the marquee by keyboard input or tilt input.

3.2 Keyboard Interaction

Because almost all mobile devices host a four-directional cross-hair control and OK/enter button as part of its keypad, we decided to optimize our small application for interaction with these five buttons. That way we could be sure that users are somewhat familiar with this control, giving us a good basis to compare with. Manipulating the selection marquee is implemented by cycling through anchor points with the centre button and moving the selected anchor with the keypad (see Fig. 2(a)).

3.3 Tilt-based Interaction

As already discussed in section 2, tilt-based interaction is so far mostly limited to panning maps or discretely “hopping” through menu or desktop items. We introduce to tilt-based control a general interaction metaphor inspired by marble maze games we call *MarbleControl*. The idea is to move a virtual ball or marble on a plane by tilting the device, and interacting with on-screen elements via the marble. This allows for one-handed interaction with a mobile device based on a simple physical simulation. By employing a model which is closely related to physical reality, the user’s approach to device interaction will be based

¹<http://www.gesturetekmobile.com/>

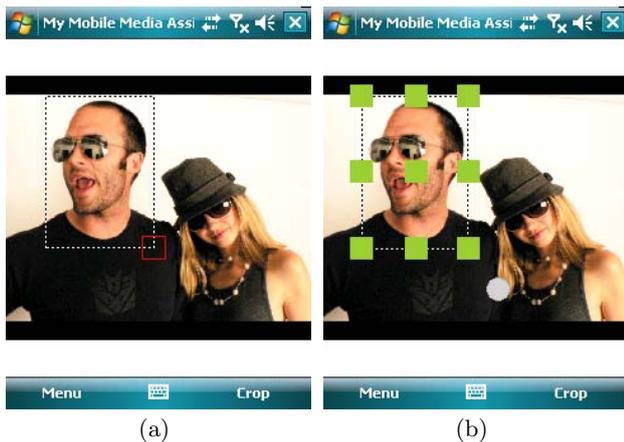


Figure 2: Screenshots of the image cropping application: The interface for keyboard interaction (a) and MarbleControl (b)

on her experience with the real world and therefore be more natural. With this we follow Donald Norman’s Execution/Evaluation Action Cycle [6].

MarbleControl works through an additional layer that is blended onto regular controls only when an activation button is pressed. This can be any easily accessible button, and is best mapped to the equivalent of the OK/enter button of the mobile device’s keypad. As soon as this happens, the marble appears as well as hot spots which the marble can interact with. Marbles and hot spots are visually easily distinguishable from the underlying controls (Fig. 2(b)).

The marble is displayed as a circle on the screen when the user presses the activation button. By pitching and rolling the device the marble is moved depending on direction and amount of tilting. To remove the accelerometer jitter a low pass filter is applied. The effective angles are calculated relatively to the starting orientation of the device when the activation button is pressed to let the marble appear. This way, a user sitting on a chair might start with a more vertically oriented device while a user standing can start with a more horizontally oriented device. In both cases, the marble will behave the same. Releasing the activation button makes the marble vanish.

Hot Spots represent active elements of the application which can be continuously manipulated, e.g. the edges of the selection marquee in an image manipulation program. When the user rolls the marble over a hot spot and releases the activation button it will stick to it. It requires a further press of the activation button to unstick the marble. In our example image editing application, we implemented a type of hot spot that moves along once the marble is stuck to it, thus making marquee movement and scaling possible (see Fig. 2(b)).

To make *MarbleControl* a realistic interaction instrument we modified the physical simulation so as to allow quite precise and direct control of the marble. First tests showed that a correct physical model of a marble on a plane was too frustrating to use due to overshoot, and didn’t allow sufficiently precise control. Thus we moved toward a simple mapping of inclination angle to the marble position. This actually moved our simulation away from (real life or digital) marble maze games, while still resembling the archetype close enough to be familiar.

4. EVALUATION

Our evaluation pitched keyboard input against tilt-based interaction via *MarbleControl* in the mobile image manipulation scenario described above. Our evaluation prototype exemplifies image manipulation by cropping photographs of people down to portrait size, which was the task given to participants.

4.1 Setup

As cropping material we prepared a series of six photographs that were put in a fixed order. The pictures had similar motifs: A shot of two persons from the waist up, one of whom was to be cropped. The photos were representative of a mobile scenario, i.e. typical shots taken with a mobile phone. We edited the pictures so that the individual head to be cropped was the same size and at the same position on every image, making each cropping essentially the same task. We then mirrored every second picture horizontally so as not to bias the left or right side.

Regarding handling of the prototype device, we examined the centre of gravity of our prototype and found that it is similar to that of a standard personal digital assistant (PDA). Despite the fact that our prototype is heavier than the used PDA alone, it rests well balanced in the user’s palm. This means there should be no negative influence on the usability of the device for the rather short duration of a trial.

Twelve people were chosen for tests and divided into two groups of three female and three male participants each. Both groups were asked to perform the cropping task for each photo, once using a key interface and once using *MarbleControl*. One group used *MarbleControl* first on each image and then used the keys to do the same, the other group did this the other way around.

4.2 Results

The average task completion time utilising *MarbleControl* was 44.6 seconds ($t_{min}=22.6s$, $t_{max}=169s$). In contrast to that the participants needed an average of 22.6 seconds ($t_{min}=15.7s$, $t_{max}=73.7s$) to complete a task using the key interface. Figure 3 shows the average task completion times for both interaction paradigms. It is clearly visible that the initially high values dropped about 50% after just two cropped pictures. Removing these learning tasks from the calculation, one gets an average of 35.4 seconds for *Marble Control* and 23.1 seconds for the key interface.

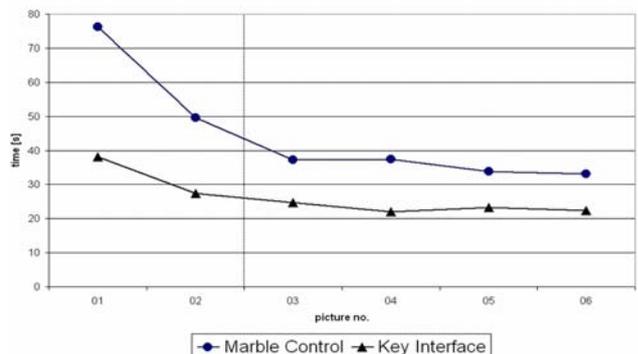


Figure 3: Average task completion times for both interaction paradigms

Nine out of twelve participants judged the key interface to be more precise than *MarbleControl*. This assessment is supported by the review of the cropped pictures which show more inaccuracy for *MarbleControl*. Nevertheless the majority of the participants stated that *MarbleControl* was more fun and that they could imagine to use it. However, some of them suggested that applications with less accuracy demanding tasks would be more adequate for this motion-based interaction method.

In some cases where the device had to be tilted to a large angle of inclination, we observed that participants encountered several problems. For one thing, people stated that they had to strain their wrists too much. Further, they were irritated by reflections at high angles of inclination obscuring their view of the display. Lastly, the marble movement seemed unrealistic to users when strongly tilting the device.

4.3 Discussion

The observed problems resulting from having to tilt the device too far can be partly explained by our direct mapping of inclination angle to marble position (see section 3.3). This allows for more precise manipulation but also requires more movement for coarse manipulation. A more physically correct model for *MarbleControl* would have a more realistic feel, but would not be easier to use due to overshoot (as showed in our pre-tests). The inconsistencies caused by the employed model for marble movement can possibly be solved by lowering the tilt range, i.e. decreasing the lowest tilt angle at which the marble touches the screen edge. This trade-off between speed and precision is hard to specify but rather needs to be determined by trial-and-error.

A further source of inconsistency was that tilt angles were mapped relative to activation tilt rather than absolute (see section 3.3). Users expected the marble to start rolling when tilting the device prior to activation of *MarbleControl*. This mainly occurred with tilting to the right or left.

5. FUTURE WORK

This short study pits one implementation of tilt-based control against a standard keyboard control. Other implementations of tilt interaction are conceivable, and should also be incorporated in an evaluation. As this sub-field of motion-based interaction is far from standardised, this will possibly require more creative work and demand for other novel metaphors such as the one described in this paper.

In future work we intend to expand and refine the *MarbleControl* metaphor and further evaluate it. It will also be interesting to explore the different possibilities to implement the marble's physical model.

It further remains to be seen how *MarbleControl* performs in other application contexts, i.e. more complex interfaces where a general pointer could outperform tabbing through controls with key presses.

6. CONCLUSION

In this paper we compared the fairly new concept of tilt-based interaction with the established keyboard interface for precise, one-handed manipulation on mobile devices. As a representative application we selected image editing, such as cutting snapshot photographs. For this we introduced a new interface metaphor for mobile device interaction we call *MarbleControl*, which is inspired by the

physical concept of a marble maze toy, where a ball or marble is moved on a plane by tilting the plane surface on two axes.

Our evaluation showed that compared to a hardware key control, *MarbleControl* is slower in the average and slightly less precise. As long as an application only has very few means of interaction and requires pixel precise control, keys seems to remain the right choice. We believe that the main strength of *MarbleControl* shows in more complex interfaces which require the user to interact with many sliders, buttons and similar control elements.

7. ACKNOWLEDGMENTS

Many thanks go to Jens Teichert for his patient help with building our hardware prototype.

8. REFERENCES

- [1] S.-J. Cho, C. Choi, Y. Sung, K. Lee, Y.-B. Kim, and R. Murray-Smith. Dynamics of tilt-based browsing on mobile devices. pages 1947–1952, San Jose, CA, USA, 2007. ACM Press.
- [2] L. Kratz, M. Smith, and F. J. Lee. Wiizards: 3d gesture recognition for game play input. pages 209–212, Toronto, Canada, 2007. ACM.
- [3] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio. Enabling fast and effortless customisation in accelerometer based gesture interaction. pages 25–31, College Park, Maryland, 2004. ACM.
- [4] V.-M. Mantyla, J. Mantyjärvi, T. Seppanen, and E. Tuulari. Hand gesture recognition of a mobile device user. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 281–284 vol.1, 2000.
- [5] M. Merdes, J. Häußler, and M. Jöst. ‘SlidingMap’: introducing and evaluating a new modality for map interaction. In R. Sharma, T. Darrell, M. P. Harper, G. Lazzari, and M. Turk, editors, *ICMI*, pages 325–326. ACM, 2004.
- [6] D. A. Norman. *The design of everyday things*. Basic Books New York, 2002.
- [7] I. Oakley and S. O’Modhrain. Tilt to scroll: evaluating a motion based vibrotactile mobile interface. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 40–49, 2005.
- [8] J. Rekimoto. Tilting operations for small screen interfaces. pages 167–168, Seattle, Washington, United States, 1996. ACM.