A Motion-based Marking Menu System

Ian Oakley
Smart Interface Research Team
Electronics and
Telecommunications Research
Institute
Daejeon, Korea
ian@etri.re.kr

Junseok Park
Smart Interface Research Team
Electronics and
Telecommunications Research
Institute
Daejeon, Korea
parkjs@etri.re.kr

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
ACM 978-1-59593-642-4/07/0004.

Abstract
The rapid development of handheld devices is driving the development of new interaction styles. This paper examines one such technique: using hand motions to control a menu system. Previous research on this topic deals with systems which rely heavily on graphical feedback, a disadvantage in many mobile scenarios. Inspired by marking menus, our system is designed to be used 'eyes-free' and based on making relatively large scale rotational strokes. We describe the system and an initial evaluation in detail. The results indicate that its performance is comparable to previous motion menu systems, but that this can be attained without visual feedback. This represents a substantial benefit.

Keywords
Mobile, motion interface, gesture, marking menus.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies.

Introduction
Mobile devices pervade our lives, supporting an ever greater number of tasks from photo-taking through browsing the internet to media-playing and communication using everything from text to voice and video. However, despite the breadth of functionality now supported, many devices still feature the same

basic physical interfaces found on the initial models. Input is entered through a limited set of buttons and the system's responses observed on a small screen. Although the devices have become more sophisticated, the basic paradigms for interaction have remained the same. This discrepancy has led to a body of research examining new and alternative approaches to mobile interaction. One technology that shows considerable promise is motion sensing. It has a number of advantages: it is rich and expressive (with 6 degrees of freedom) and naturally accessible in the context of handheld mobile devices. Furthermore, the sensors needed to monitor it are small, cheap, power efficient and require no real-estate on the casing of a device.

A number of authors have investigated this modality and one topic that has attracted interest is deep menu systems (as these are the dominant interface on mobile phones). Poupyrev *et al.* [3] and Oakley and O'Modhrain [1] both describe and evaluate menu systems controlled by varying the orientation of a handheld device. Poupyrev's work maps changes in orientation to the speed at which menu items are traversed, while Oakley's involves associating fixed segments of orientation with individual menu items. The first metaphor has the advantage that it can be employed with menus of arbitrary length, the second that it is conceptually simple and may be able to better capitalize on the kinesthetic memory of users.

However, these designs both share a basic reliance on interactive graphical feedback, something which is ill-suited for motion interfaces. In both these systems, and despite the fact that these two acts clearly conflict with one another, users need to simultaneously move the device and observe its screen, a challenging task.

TiltText [6], a motion interface supporting text-entry, arguably owes its high levels of performance to the fact that it relies on movements that can be performed unmonitored. In fact, as Pirhonen *et al.* [2] point out, interfaces that do not rely on graphical feedback ('eyes-free' interfaces) are especially well suited to many mobile tasks. Indeed, this is one reason why physical buttons remain an enduring feature. With a familiar mobile device, a user can find and activate a button just by touch, and such solely haptic interaction is used in many commonplace scenarios such as answering calls, halting alarms and adjusting volume. Responding to these issues, this paper presents the design and initial evaluation of a motion based menu system intended to be used 'eyes-free'. We believe that adopting such an approach is an important step in popularizing motion based interfaces and enabling their transition from the sterile lab to the messier, noisier, more complex environments of the real world.

Motion Marking Menu System

The menu system proposed in this paper is derived from recent work on marking menus [7], a simple form of gestural interface often used with styli, but rarely employed in other contexts. Gestural interaction paradigms are a good fit for the fluid, continuous and unmonitored style of normal human motion and one of the key benefits of marking menus is that they seamlessly support the transition between novices and experts. They feature full graphical interfaces for novices, but are designed such that regular use serves to produce experts who are no longer reliant on the visual feedback. This trait sets them apart from the majority of gestural systems which require substantial training for either the users (to remember commands) or the system (to recognize them).

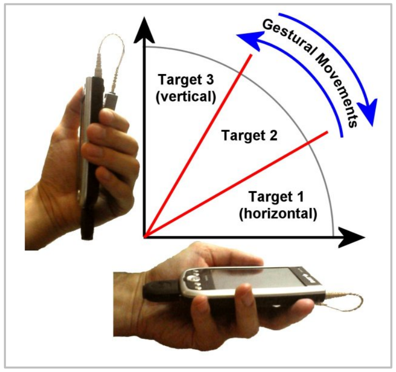

Figure 1. Illustration of the three target gestural input system. To use the system, the device is rotated between horizontal and vertical positions.

The system involves dividing a 90 degree portion of rotational space (from holding a device horizontally to holding it vertically) into three targets, as shown in Figure 1. Commands are issued by rotating a handheld device into one of the targets, and pressing a button (or against a touch screen). Releasing the button immediately issues what we term a no-stroke command. Rotating the device into another target (and therefore making a stroke or mark) before releasing the button results in a one-stroke command. A two-stroke command can be produced by changing the direction of the rotation (at the level of targets) and performing another mark. Additional strokes can be added the same way, at the cost of increasing the complexity of required movements. The first three command types are conceptually illustrated in Figure 2, while Figure 3 sketches how this system might be realized on a device.

This design diverges from traditional marking menus in several ways. Firstly, it is one-dimensional. Secondly, to compensate for the reduction in expressiveness this causes, bounded strokes (in which both the start and end point contribute meaning) are used. These choices reflect the observation that it is more difficult to accurately control tilt in two dimensions than in one. One key reason for this is that tilt interfaces operate relative to gravity: it is clear how to rotate a device when it is horizontal (its axes aligned to gravity), but independently adjusting one axis when the other is at 60 degrees is much more complex. In this initial work, we chose to use a single axis to sidestep this issue.

Hardware and Software Platform

Our system ran on a 624 MHz Dell Axim X51v under MS Windows Mobile Version 5. To capture movements we used the TiltCONTROL, a sub-$100 device built by PocketMotion [4]. It features a 2-axis accelerometer packaged with a microprocessor which provides an RS232 interface (some rewiring was required to make it compatible with the X51's serial connector). A simple API allows it to be easily integrated into an application, and functions exist to provide orientation (filtered with a simple 8 sample rolling average algorithm). The previous literature suggests vibrotactile cues are an effective way to support motion interfaces [1, 3], so we integrated these into our system using a VBW32 skin transducer from Audiological Engineering [5]. We attached the transducer to the back of the PDA with a Velcro strip, and drove it from the PDA's headphone jack. Our final device is illustrated in Figure 4.

Study

The goals of this initial study were to gather basic data to compare against the literature and to assess the feasibility of the 'eyes-free' performance of the system. 8 employees from our institute took part, 4 were male, 4 female, all right handed. Their mean age was 28.

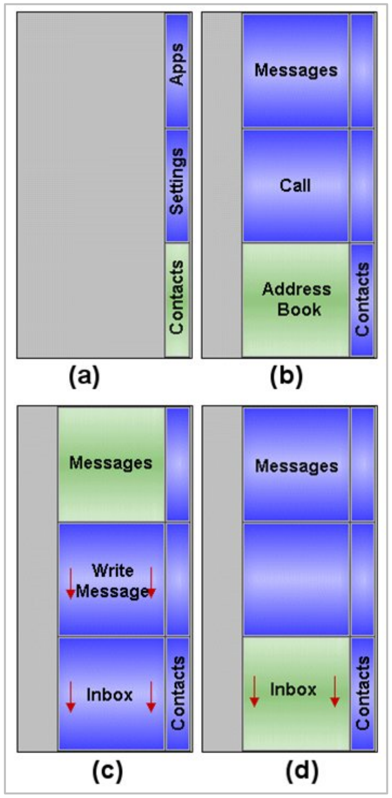

Figure 3. Four screenshots of the proposed UI. Icons at the bottom signify a horizontal orientation, icons at the top, a vertical one. In (a) the user is holding the device horizontally, in (b) they have activated the menu system, in (c) the device has been rotated to vertical and in (d) returned to a horizontal orientation. Releasing the screen will then activate the highlighted command (to view the Inbox).

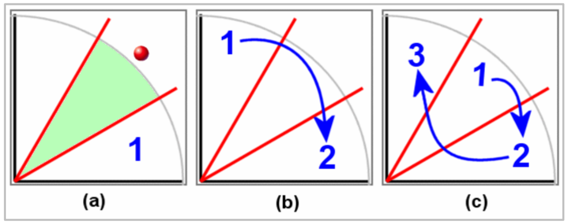

Figure 2. Three images of the experimental interface showing gesture commands: a no-stroke command (a), a one-stroke command (b) and a two-stroke command (c). The numbers and arrows indicate the path of each of the gestures. For instance, (c) shows a gesture which starts in the central target, involves a rotation to the horizontal one, and a second rotation to the vertical one. Image (a) also shows the highlighting used in the Visual condition: the currently occupied target is a darker color, while the presence of the round icon indicates a gesture is underway.

Materials

The menu system used in this study featured three targets, and we considered commands composed of up to two strokes. This resulted in a menu system capable of issuing 19 separate commands. Two target sizes were used: 30 and 45 degrees. However, subsequent analysis revealed no performance differences between them so, in the interests of brevity, no further details are included on this aspect of the study in this paper. Figure 2 shows examples of the experimental interface. Numbers and arrows instruct users as to the path they should take, while graphical feedback in the form of darker green highlighting (signifying the current target) and a small round red icon (signifying a gesture is in progress) could be enabled or disabled to indicate or hide state. This interface was designed solely to convey the required movements simply, as the goal of this initial study was to assess whether or not users could learn the required motions and not the usability of a particular interface (Figure 3 is a possible interface).

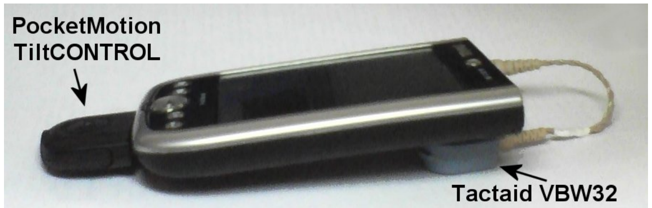

Figure 4. Hardware used to develop interface featuring motion sensor and vibrotactile display device.

Targets dynamically adjusted their size when users moved between them: the source target shrank by 2 degrees while the destination target expanded by a similar amount. This was designed to minimize accidental transitions. Tactile cues were also displayed to a user as they moved between targets. We used a 100ms sample composed of a 250Hz waveform (the VBW32's resonant frequency) with a curved amplitude envelope. It resembled the feel of a brief click.

Experimental Design and Measures

This study had 2 Visual blocks each with 114 trials (each menu command, 6 times) and involving the display of the graphical feedback. This stage was intended to enable users to attain some expertise with the system. Subjects then completed a Blind condition composed of 38 trials (each menu item, twice) in which graphical feedback was absent. Half-sized practice blocks were presented prior to each experimental block and all trials were delivered in a random order. The experimental measures were task time and error rate.

Procedures

The experiment took place in an unused office. All participants stood and used the PDA one-handed in their dominant hand. Gestures were initiated and terminated by pressing and releasing anywhere on the touch screen. The interface was shown on a laptop in front of the user (controlled by the PDA via Bluetooth). Each trial began by instructing users to take a brief rest and to tap the PDA screen to proceed. When they did so, a fixation spot was displayed for 500 ms, followed by the experimental interface (as described in the materials section). On the appearance of this screen, a user's task was to rotate the PDA to the first target, press against its screen and then (if required) perform further rotations until releasing the screen upon arrival at the final target. At this stage, a new trial began. Rest breaks between the conditions were also enforced.

Results and Discussion

The timing and error data are presented in Figures 5 and 6. They include means, and a breakdown of the data into trials with different numbers of strokes. T-tests on the means showed the Visual condition led to more rapid task times than the Blind ($p<0.001$), but no difference was found for error rate ($p=0.1$).

A repeated measures ANOVA used to analyze the data according to the number of strokes in each trial showed clear effects in task time ($F(2,23)=155$, $p<0.001$) and error rate ($F(2,23)=10.7$, $p<0.001$). *Post hoc* pair-wise comparisons for task time were all highly significant ($p<0.001$) while no-stroke trials resulted in fewer errors than one and two-stroke trials ($p<0.001$). Examining these results with the raw data suggests that making additional strokes exerts an (at worst) linearly increasing time cost to the task and that, after the first stroke, error rates remain flat (at least in the Visual condition). These observations indicate that gestures of several strokes are no more difficult to make than those of a single stroke and are supportive of the designs ability to scale up to include more stokes.

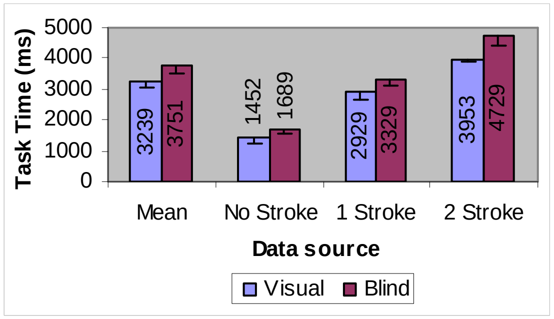

Figure 5. Task times from study (bars show standard error).

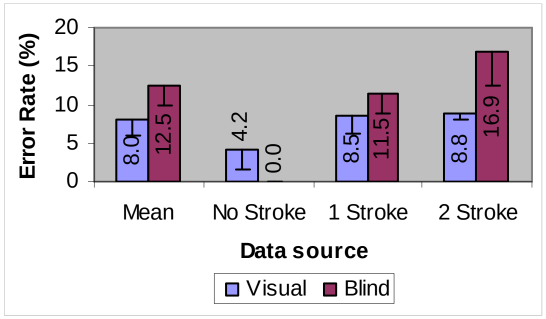

Figure 6. Error rates from study (bars show standard error)

However, the primary conclusion from this study is that there is modest but clear dip in performance in the Blind condition; task times increase, but remain respectable, while error rates do not change significantly. Indeed, the temporal increase may simply be attributable to the participant's relatively short experience with the system – a weakness common in the lab-based evaluation of gestural interfaces.

Contrasting these figures with the literature on motion controlled menus is informative. Poupyrev [3] reports a task time of 3.1 to 3.7 seconds for menus items 6 and

12 items distant; 3.4 seconds is a likely extrapolation to a 19 item menu. No error rates are given, but Oakley & O'Modhrain [1] re-ran this study and reported a rate of 19% for a 15 item menu. Hence, Poupyrev's temporal results are broadly similar to those attained in this paper, but the error rate is considerably worse. Oakley's own technique [1] leads to a 2.75 second task time and an error rate of nearly 10% with a 15 item menu; this technique is faster than the one presented here, but has an error rate squarely between those of the Visual and Blind conditions. We suggest that both these differences are probably due to the larger target sizes we employ: crossing larger targets incurs a time cost but confers an improvement in error rate (at least in the Visual condition).

In summary, this review places the technique proposed in this paper in the middle of the performance bracket outlined by the previous literature. However, we feel the most important aspect of this work is represented by the Blind condition. This study shows that, after a short training period, the system is usable without graphical feedback, something not remotely possible (or even considered) with previous motion based menu systems, and which dramatically changes the manner in which it can be used. Rather than requiring continual visual monitoring, our system is designed to support motions that can be internalized, learnt and issued semi-automatically, by feel alone. The results of this study suggest this goal can be achieved, and are supportive of additional work towards this.

Conclusions

We have presented a novel input system based on motion sensing and marking menus. We discuss its design then provide an initial evaluation demonstrating that it can be used without graphical feedback. Future work will include more practically based empirical studies featuring a realistic user interface and further investigations to characterize user performance with different numbers of targets and in comparison to alternative input technologies such as keys or joysticks. We are also interested in exploring a two-dimensional ballistic version, and looking at learning effects and more expert users. We believe that motion based interfaces have a role to play in the next generation of mobile devices, and that designs which aim to be 'eyes free' will significantly advance their adoption by offering a better fit with real world mobile device use.

Acknowledgements

This work was supported by the Korean Ministry of Information and Communication. Thanks must go to Kiuk Kyung, Jieun Park and Youngkyu Jung for their help with electronics, translation and mathematics.

References

[1] Oakley, I. & O'Modhrain, S. Tilt to Scroll. In *Proc. of World Haptics'05*, Pisa, Italy (2005).

[2] Pirhonen, A. Brewster, S.A. & Holguin, C. Gestural and Audio Metaphors as a Means of Control for Mobile Devices. In *Proc. Of CHI 2002*, ACM Press (2002).

[3] Poupyrev, I., Maruyama, S. and Rekimoto, J. Ambient touch: designing tactile interfaces for handheld devices. In *Proc. of ACM UIST 2002*, ACM Press (2002).

[4] PocketMotion TiltControl. www.pocketmotion.com.

[5] Tactaid VBW. www.tactaid.com/skinstimulator.html.

[6] Wigdor, D., & Balakrishnan, R. (2003). TiltText: Using tilt for text input to mobile phones. In *Proc. of ACM UIST 2003*, ACM Press (2003).

[7] Zhao, S., Agrawala, M., Hinckley, K., Baudisch, P. Zone and Polygon Menus. In *Proc. of ACM CHI 2006*.