ARC-Pad: Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen

David C. McCallum
University of Manitoba
Computer Science Dept, E2-560 EITC
Winnipeg, Manitoba R3T 2N2
ummccal1@cc.umanitoba.ca

Pourang Irani
University of Manitoba
Computer Science Dept, E2-580 EITC
Winnipeg, Manitoba R3T 2N2
irani@cs.umanitoba.ca

Figure 1: (Left) Cursor is initially at the top right corner. (Center) Tapping anywhere with ARC-Pad causes the cursor to instantly jump across the screen. (Right) For accurate positioning the user can clutch and slide the finger.

ABSTRACT

We introduce ARC-Pad (Absolute+Relative Cursor pad), a novel technique for interacting with large displays using a mobile phone's touchscreen. In ARC-Pad we combine absolute and relative cursor positioning. Tapping with ARC-Pad causes the cursor to jump to the corresponding location on the screen, providing rapid movement across large distances. For fine position control, users can also clutch using relative mode. Unlike prior hybrid cursor positioning techniques, ARC-Pad does not require an explicit switch between relative and absolute modes. We compared ARC-Pad with the relative positioning commonly found on touchpads. Users were given a target acquisition task on a large display, and results showed that they were faster with ARC-Pad, without sacrificing accuracy. Users welcomed the benefits associated with ARC-Pad.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces – Input devices and strategies

General terms: Design, Human Factors

Keywords: Touchpad, cursor, clutching, absolute position

UIST'09, October 4–7, 2009, Victoria, British Columbia, Canada.

INTRODUCTION

We are witnessing a recent trend involving the use of mobile devices to control and manipulate objects, such as displacing a cursor, on a secondary peripheral display. Some methods include tilting the device, such as the Air Mouse [10], or controlling objects using a phone-cam [7]. In this paper, we propose an alternative, and focus on methods of using the device's touchscreen as a touchpad.

Touchpads have a very small size compared to the user's primary display and require clutching to move the cursor. Clutching degrades performance [3], particularly when the display size is very large. A simple solution to minimize clutching involves increasing the cursor speed with a multiplier called the Control-Display gain (CD gain). CD gain amplifies the fingers' movements, so that small motions on the touchpad result in large movements of the cursor. However, increasing the CD gain reduces accuracy, making smaller objects more difficult to target [2]. Alternatively, researchers have proposed using hybrid techniques to eliminate or reduce clutching.

Building on prior techniques we introduce ARC-Pad, which is intended to increase selection speed without reducing accuracy. ARC-Pad does not require explicit switching between input modes as with prior techniques [4]. When the user taps and releases, the cursor momentarily switches to absolute mode, and jumps to the equivalent position on the screen. Any gesture where the user's finger slides invokes relative motion. For example, a tap in the lower left corner of the touchpad instantly moves the cursor to the lower left of the screen (Figure 1). Users can jump to travel long distances, and use relative motion for more accurate positioning. In a target acquisition task users were faster with ARC-Pad than relative control, without losing accuracy.

RELATED WORK

We briefly present techniques for direct long range interaction, improvements to touchpad input and finally hybrid relative-absolute techniques.

Researchers have proposed numerous techniques for interacting with large displays [15, 6, 12, 1, 11, 14]. Users can directly interact with the physical space [15]. Techniques exist for extending the cursor (drag-and-throw) [6], for pointing at remote objects with a variation on the cursor [12], or bringing objects closer to the cursor [1]. Alternatively, users can position a cursor with a mini-map of the workspace, such as in radar views [11]. One issue with radar, however, is that a mode switch is normally required to activate the view, and it occludes items behind it.

Various solutions exist for improving touchpad performance. Cursor acceleration is commonly seen on many operating systems [13]. With cursor acceleration, the CD gain is dynamically adjusted based on the velocity of the cursor movement. Rapid cursor movements increase the CD gain. Cursor acceleration has to be balanced, since high levels of cursor acceleration can cause users to overshoot targets [2].

RubberEdge and EdgeMotion are hybrid systems that use position control when the user's finger is at the center of the touchpad, and switch to rate control when the finger is at the edge [3]. RubberEdge has been shown to outperform cursor acceleration under various contexts [3].

Precise and Rapid Interaction through Scaled Manipulation (PRISM) [5] dynamically adjusts CD gain in a 3D environment with direct input. When the CD gain is 1, the item being manipulated follows the user's hand. When the CD gain is reduced, the item moves more slowly. PRISM is a hybrid system, since it has relative and absolute positioning.

HybridPointing allows users to control the input either in absolute or relative mode, and was designed to minimize the large distances that users must cover when interacting with large wall displays [4]. With this technique all input is direct, by touching a stylus to the display. The user has to explicitly switch between absolute and relative control. A circular trailing widget follows the cursor, and users tap it to switch to relative mode. Note that while the input is direct, the cursor is not necessarily beneath the stylus when relative motion is used. Users lift their stylus from the screen to switch back to absolute mode. Participants were slower when using HybridPointing than with relative positioning alone, even if they didn't transition to absolute mode. The authors suggested it may be due to the cognitive load of managing multiple modes.

DESIGN GOALS FOR ARC-PAD

What seems prevalent in prior work on touchpads is a need to balance CD gain to suit the appropriate targeting conditions. Ideally, one technique should be able to operate well under most, if not all conditions. This was the primary motivating objective for the design of ARC-Pad. We identified two design goals to achieve this objective.

Reduce Cognitive Load

ARC-Pad is inspired from the design of previous hybrid techniques, such as HybridPointing [4]. However, studies have inferred that switching modes can increase cognitive overhead and decrease performance. In the design of ARC-Pad we imposed a criterion to reduce explicit mode switching between absolute and relative control. Users must decide whether to perform an absolute jump, but have no need to switch back to relative mode afterwards.

Reduce Occlusion of the Screen

Prior techniques, including radar views [14], occlude part of the display. Direct input can also occlude regions of the display. However, when using mobile devices, it is possible that several users could be collaborating with the display simultaneously. Indirect input in this case is ideal, since the user's hands aren't overlapping the display.

IMPLEMENTATION

The Absolute+Relative Cursor Pad (ARC-Pad) is a hybrid positioning system for small touchpads. It allows absolute and relative positioning. Locations on the touchpad are mapped to a larger display. A quick tap and release causes the cursor to instantly jump to the related location on the screen. To invoke relative mode, the user needs to slide the finger along the touchpad, similar to working with conventional touchpads. ARC-Pad uses cursor acceleration in relative mode. Since the tap gesture is taken for jumping, clicking is performed by pressing a separate selection button with the thumb of the opposite hand. We implemented ARC-Pad on an HTC TouchDiamond mobile phone with a 640×480 display resolution and a physical button using a finite state machine (Figure 2).

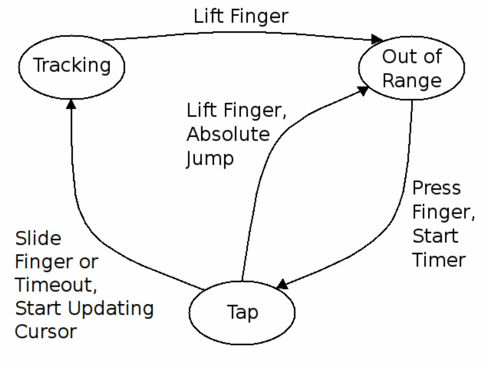

Figure 2: The ARC-Pad finite state machine.

ARC-Pad is initially in the Out of Range state. When the screen is touched, it moves into the Tap state. The user can either lift her finger to perform an absolute jump, or slide along the surface to engage relative motion. The threshold value for detecting a finger slide was set to 20 pixels on the phone surface, and any movement less than this amount would not register as a slide. The user must tap and release without sliding within 300 milliseconds to perform a jump. Otherwise, the system enters the tracking state. These restrictions reduce accidental jumps from casually touching the surface. Clutching is provided as normal. It can be difficult to jump to the very edge of the screen, but we can get sufficiently close to select targets without extra clutching.

EXPERIMENT

In this experiment we examined ARC-Pad's effectiveness.

Method

Nine participants (7 male, 2 female) volunteered for the experiment. We used a serial target acquisition task similar to Forlines et al. [4], consisting of selecting square targets on a large screen using the touchpad. We measured completion time to select the targets with the following conditions:

- Technique type: ARC-Pad, Cursor Acceleration (CA).
- Distance: 625, 937, and 1250 pixels.
- Target Size: 9, 15, and 21 pixels.

Prior to starting the experiment, participants carried out 10 practice trials. Upon completing the practice trials, participants then performed 30 trials per condition, for a total of 180 trials per technique type. A trial ended when the participant correctly selected the target. We counterbalanced the presentation based on technique. Levels of the remaining factors (distance, size) were presented randomly. We explained the task and each technique, and participants were given 20 practice trials for each technique. We used Windows XP's default cursor acceleration curve for ARC-Pad and CA [13]. CD-gain at low speeds reached 2.06 and at high speed was at 9.1. The range, minimum and maximum values are larger than those tested by Casiez et al. [3], providing faster cursor movement. We did not compare ARC-Pad to PRISM as it uses direct input in a 3D environment, whereas CA is designed for indirect input with a touchpad.

Materials

The primary screen was a 52" monitor with a resolution of 1360×768 and 1.2 pixels/mm. We simulated a touchpad using an HTC touch smartphone as our primary device as it had a responsive touch. The resolution of the HTC screen was 640×480. Since the aspect ratios of the two screens did not match, a horizontal movement of 1 pixel on the phone corresponded to 0.66 pixels on the screen, and a vertical movement of 1 pixel mapped to 0.54 pixels on the screen.

Results

We used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) for all our analyses with subjects as random factor.

Completion time

Average completion for ARC was 2623ms (s.e. 13.8) and for Cursor Acceleration was 2789ms (s.e. 13.8). There was a significant effect for *technique* ($F_{1,8}=7.67$, $p=0.024$), of *distance* ($F_{2,16}=55.1$, $p<0.001$), and for *size* ($F_{2,16}=97.74$, $p<0.001$) on completion time. We found a significant interaction effect for *technique*×*distance* ($F_{2,16}=3.67$, $p<0.048$) but not for *technique*×*size* ($F_{2,16}=0.4$, $p=0.678$). There was no interaction effect for *distance*×*size* ($F_{4,32}=2.27$, $p=0.084$). Figure 3 (left) shows average completion time for technique by distance.

Post-hoc pair-wise comparisons (unequal variance assumed) yielded significant differences (all $p<0.001$) in trial completion times for all pairs of *distance* and *size*.

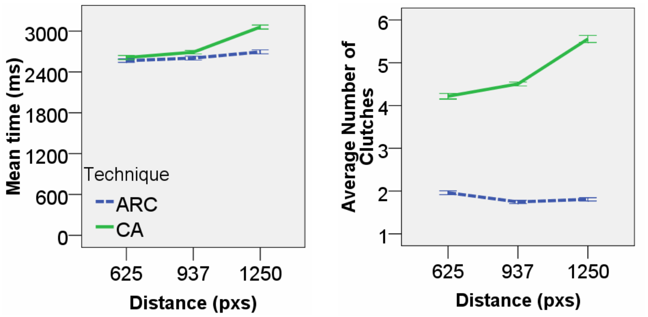

Figure 3: (left) Average completion time; (right) average number of clutches, for both techniques by distance. (bars represent 1 s.e.)

ARC-Pad outperformed our CA settings, particularly after the 937 pixel mark, suggesting that distances beyond this value would result in larger improvements.

Number of clutches

We recorded the number of clutches performed in both techniques. The HTC TouchDiamond is not as sensitive as a conventional touchpad and would occasionally lose track of the user's finger, causing unwanted absolute jumps. We counted all absolute jumps as clutches and added any subsequent clutches incurred by the technique.

ARC-Pad resulted in an average of 1.84 (s.e. .026) clutches and Cursor Acceleration produced an average of 4.76 (s.e. .026) clutches. There was a significant effect for *technique* ($F_{1,8}=155.74$, $p<0.001$), for *distance* ($F_{2,16}=10.98$, $p<0.001$), and for *size* ($F_{2,16}=3.63$, $p=0.05$) on number of clutches. We found a significant interaction effect for *technique*×*distance* ($F_{2,16}=12.37$, $p<0.001$) but not for *technique*×*size* ($F_{2,16}=0.014$, $p=0.986$). There was no interaction effect for *distance*×*size* ($F_{4,32}=1.516$, $p=0.221$). Figure 3 (left) shows average completion time for technique by *size* and *distance*.

The results suggest that across all distances, ARC-Pad is able to reduce the amount of clutching by one half. This result corroborates the findings of Casiez et al. [3].

User preferences

After the experiment participants rated the techniques. Over half preferred ARC-Pad. The remainder mostly remarked that ARC-Pad required a novel and unfamiliar shift in the manner of interacting with a touchpad. Most participants said they would use such a technique on their system. Due to screen sensitivity on our device, we witnessed occasional unwanted absolute jumps. A more sensitive device would remove this limitation. Visually searching for the cursor after jumping was not difficult since the cursor was still in motion from finger contact on the device.

APPLICATIONS AND FUTURE WORK

Our experiment demonstrated the effectiveness of a low-cost absolute and relative pointing system for touchpads. In this section, we describe applications that can benefit from extensions of ARC-Pad.

Absolute-Drag

Dragging objects over large distances can be frustrating with touchpad-style interactions. Dragging can be easily carried out in our current implementation. The user presses and holds the selection button to drag the object, and taps to invoke an absolute jump. The object jumps along with the cursor. When the button is released, the object is dropped at the new position. We eliminate dragging and reduce the operation to a select-tap-and-release. Absolute–drag works transparently in Windows.

Including a Radar View

We can augment ARC-Pad by allowing the user to see parts of the large display on the cellphone. The system would stream screenshots to the cellphone screen, and guide the user's absolute tapping, similar to radar views [14].

Touchpad Interactions and Multi-Display Cursor

ARC-Pad uses touchpad-based metaphors, and mobile devices such as laptops that have touchpads could benefit from ARC-Pad. Current touchpads (synaptics.com) provide absolute coordinates, and it would be trivial to extend their behavior to allow hybrid operation.

Our results show that ARC-Pad is especially useful for long range target acquisition. Often laptop users will connect an external monitor to extend the visible workspace. To switch between the two displays, users typically need to move the cursor from one screen to the other, causing several problems such as losing the cursor or travelling long distances between displays. Researchers have proposed a number of solutions, such as replicating the cursor [8]. ARC-Pad can easily be extended such that one half of the pad is dedicated to one screen. Tapping in that region causes the cursor to jump to the intended screen. Additionally, if the workspace were visually presented to the user, each workspace could be represented as a separate thumbnail on the touchscreen.

CONCLUSION

Our primary contribution includes ARC-Pad, a novel technique for interacting on large displays with a conventional touch-enabled mobile device, such as a cell-phone. ARC-Pad allows instantaneous jumps through absolute positioning and lets users refine their pointing with relative cursor movement. Our study shows that users are faster at selecting targets with ARC-Pad than with a cursor acceleration technique, particularly when targets are far away from the cursor. ARC-Pad's absolute positioning reduces clutching by half. In our future work we will add a mini-map of the larger workspace to guide absolute jumping.

REFERENCES

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: techniques for accessing remote screen content on touch- and pen-operated systems, *Proc. of Interact 2003*, 57-64.
2. Casiez, G., Vogel, D., Balakrishnan, R. and Cockburn, A. The impact of control-display gain on user performance in pointing tasks, *Human-Computer Interaction 2008*, Volume 23, 215-250.
3. Casiez, G., Vogel, D., Pan, Q. and Chaillou, C. Rubber-Edge: reducing clutching by combining position and rate control with elastic feedback. *Proc. of UIST 2007,* 129-138.
4. Forlines, C., Vogel, D., Balakrishnan, R. Hybridpointing: Fluid switching between absolute and relative pointing with a direct input device, *Proc. of UIST 2006*, 211-220.
5. Frees, S., Kessler, G. D., and Kay, E. PRISM Interaction for enhancing control in immersive virtual environments, *Proc of TOCHI 2007*, Volume 14, Article 2
6. Hascoët, M. Throwing models for large displays. In *Proc. of HCI΄2003*, British HCI Group, 73-77.
7. Jiang, H., Ofek, E., Moraveji, N. and Shi, Y. Direct Pointer: Direct manipulation for large-display interaction using handheld cameras. *Proc. of CHI 2006*, 1107-1110.
8. Kobayashi, M. and Igarashi, T. Ninja cursors: using multiple cursors to assist target acquisition on large screens, *Proc. of CHI 2008*, 949-958.
9. MacKenzie, I. S., Kauppinen, T., & Silfverberg, M. Accuracy measures for evaluating computer pointing devices, *Proc. of CHI 2001*, 9-16.
10. Mobile Air Mouse, www.mobileairmouse.com
11. Nacenta, M., Aliakseyeu, D., Subramanian, S., and Gutwin, C. A comparison of techniques for multi-display reaching, *Proc. of CHI 2005*, 371-380.
12. Parker, J., Mandryk, R., and Inkpen, K.. TractorBeam: seamless integration of local and remote pointing for tabletop displays, *Proc. of GI 2005*, 33-40.
13. Pointer ballistics for Windows XP, www.microsoft.com/whdc/archive/pointer-bal.mspx
14. Reetz, A., Gutwin, C., Stach, T., Nacenta, M., and Subramanian, S. Superflick: a natural and efficient technique for long-distance object placement on digital tables, *Proc. of GI 2006*, 163-170.
15. Rekimoto, J. Pick-and-Drop: a direct manipulation technique for multiple computer environments, *Proc of UIST 1997*, 31-39.