High Precision Touch Screen Interaction

Pär-Anders Albinsson *¶
paalb@foi.se

Shumin Zhai §¶
zhai@almaden.ibm.com

* Swedish Defence Research Agency, Division of Command and Control Systems Linköping, Sweden

§ IBM
Almaden Research Center
650 Harry Road, San Jose, CA
95120 USA

¶ Department of Computer and Information Science
Linköping University
Linköping, Sweden

ABSTRACT

Bare hand pointing on touch screens both benefits and suffers from the nature of direct input. This work explores techniques to overcome its limitations. Our goal is to design interaction tools allowing pixel level pointing in a fast and efficient manner. Based on several cycles of iterative design and testing, we propose two techniques: *Cross-Keys* that uses discrete taps on virtual keys integrated with a crosshair cursor, and an analog *Precision-Handle* that uses a leverage (gain) effect to amplify movement precision from the user's finger tip to the end cursor. We conducted a formal experiment with these two techniques, in addition to the previously known *Zoom-Pointing* and *Take-Off* as baseline anchors. Both subjective and performance measurements indicate that *Precision-Handle* and *Cross-Keys* complement existing techniques for touch screen interaction.

Categories & Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces — Input devices and strategies, Interaction styles, Evaluation /methodology

General Terms: Experimentation, Performance

Keywords: Touch screens

INTRODUCTION

Interaction on touch sensitive screens is literally the most "direct" form of HCI, where information display and control are but one surface. The zero displacement between input and output, control and feedback, hand action and eye gaze, makes touch screens very intuitive to use, particularly for novice users. Not surprisingly, touch screens have been widely and successfully used in public information kiosks, ticketing machines, bank teller machines and the like.

Besides its directness, touch screen interfaces also have additional advantages. First, because its control surface is overlaid on the display, no extra input control device or space is necessary for touch screen interaction. Second, touch screens are more robust than free moving input devices such as the mouse. This is especially true in moving, weightless or otherwise demanding environments such as those of a command and control vehicle or in space.

Being direct between control and display, touch screens also have special limitations. First, the user's finger, hand and arm can obscure part of the screen. Second, the human finger as a pointing device has very low "resolution". It is difficult to point at targets that are smaller than the finger width.

These limitations have been realized and tackled before, mostly notably by Sears, Shneiderman and colleagues [16, 17, 13]. Their basic technique, called *Take-Off*, provides a cursor above the user's finger tip with a fixed offset when touching the screen. The user drags the cursor to a desired target and lifts the finger (takes off) to select the target objects. They achieved considerable success with this technique for targets between finger size and 4 pixels. For very small targets (1 and 2 pixel targets), however, users tended to make a large amount of errors with *Take-Off*. To handle small targets, Potter and colleagues [13] used techniques relying on the system's knowledge of target locations, which essentially avoided the need of precise pointing. However, there are many situations where the system cannot know what objects are users' targets.

Instead of using a bare finger, in some cases the user may use a stylus (pen) to interact with touch screens. A stylus is a much "sharper" pointer than a finger tip, but its resolution may still not be as good as a mouse cursor. Ren and Moriya [14] investigated different strategies for handling small targets and reported that 1.8 mm (5 pixels) was a crucial limit beyond which special needs arise.

As touch screen technology becomes more available at a lower price and better quality, we expect its greater use in many different domains. We set out to explore touch screen interaction techniques that can handle pointing at individual pixel levels. High precision interaction on touch screens is necessary and important in many situations including dealing with geographical systems or high

CHI 2003, April 5–10, 2003, Ft. Lauderdale, Florida, USA.

precision drawings. One area in particular is command and control where many characteristics of the touch screen mentioned earlier are desirable, but where high accuracy techniques have to be developed in order to deal with geographical information. For example, computer supported command and control systems used in military vehicles are constrained by space limitations and rugged environments. Screen size is therefore limited. To interact with these systems—for example when deploying geographical orders—users need to maintain an overview of an area of interest (which determines the zoom level) and yet be able to point at precise locations.

In the rest of the paper we first report our iterative exploration of observe – design – implement – and (informal) test, resulting in various techniques for high precision pointing. We then study four techniques in a formal experimental user study with two techniques selected from the design exploration and two from prior art. The pros and cons of these techniques informed by the experimental results are discussed at the end of the paper.

We have made the software prototype and a video available online should the reader need to gain first hand experience with the various designs presented hereafter. (`http://www.mind.foi.se/touch`).

DESIGN EXPLORATION

The State of the Art – Zooming and Cursor Keys

The first method we explored is zooming. Since the apparent size of targets depends on the scale they are displayed in, it is always possible to use zooming to enlarge the information space to a scale in which one can comfortably point at the target with a bare finger. Indeed this is one technique commonly used today. The efficiency of using zooming as an intermediate step to pointing depends on the design and implementation of the zooming user interface. In both research [e.g. 3, 8] and commercial products, a variety of methods and devices have been used for zooming. We implemented some of the most common methods, and found that an efficient approach for the purpose of pointing is what can be called end-effect-based zooming. Commonly known as *bounding box zoom* or *marquee zoom* it is previously seen in commercial products such as Adobe Photoshop. With this approach, the user first activates a zooming mode, and then draws a bounding box that encloses the area to be zoomed in. This is done by putting down the finger for one desired corner of the box and then dragging the finger to the desired opposite corner. The resulting box is shown during this operation. When the user releases the box, the system zooms and pans to the selected area and the mode switches back to pointing. Thereby the user can point at the target in an easy scale (Figure 1).

Zooming is a powerful mechanism of interacting with multi-scale information spaces [6, 7] when one needs to change both visual and motor control scale. However, it has a fundamental drawback when used for the sole purpose of pointing whose limitation is in control, not visual resolution. When zoomed to a sub area of interest, one loses the contextual global view that can be important for the user's task. This is especially true for geographical tasks in command and control, where commanders often carry out several tasks in parallel while using the computer system. Using zooming as a pointing technique may also make pointing an unnecessarily more complex task. For example, drawing a precise line from one dot (e.g. an island in the ocean) to another (e.g. an airport on a continent) on a digital map requires the user to zoom in and click on one end, zoom out to find the other end, zoom and pan to click on the other end, then zoom out again to look at both ends, all of which can be cumbersome. It is conceivable that a separate overview window could be helpful in maintaining context in using zooming user interfaces. However, research has not found reliable performance improvement with such a solution, suggesting it is difficult to integrate information in overview and detail windows [9].

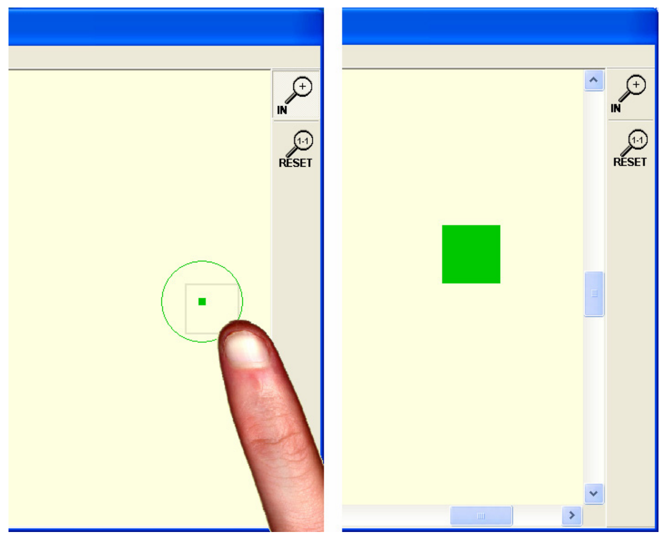

Figure 1. *Zoom-Pointing*: First the user zooms to a sub-area defined by drawing a rectangle (left). The user can then perform direct pointing at a finer scale (right). The figure shows a square target, highlighted by a surrounding circle.

Another method commonly employed for precision pointing on touch screens is to use additional cursor keys to adjust the cursor position pixel by pixel. This solution is robust but it compromises some of the basic advantages of touch screens mentioned earlier—directness and compactness. Cursor keys not only operate indirectly but also require additional control area besides the touch screen.

Amplifying Control Precision Only – *Cross-Lever*

In search of effective alternatives, we designed our first new technique: *Cross-Lever*. The goal was to amplify the control precision scale without having to change the display scale. As shown in Figure 2, *Cross-Lever* presents two crossed lines when the user first taps on the screen. The intersection between these lines indicates the point to be selected, which can be controlled by moving the two "rubber-band" lines separately. Selection is done by tapping within a certain range of the intersection point, represented by a circle. Each end of the rubber-band lines has a handle that can be dragged, making the line longer or shorter. Making the line longer will result in higher *precision leverage* to the intersecting point. By initially putting the intersecting point in an asymmetrical position the user is given the choice of using either a low precision leverage but high movement efficiency handle or vice versa. Both handles and lines are semi-transparent so as not to occlude the background.

Our informal tests quickly revealed the problems of *Cross-Lever*. Although it succeeded in allowing users to select one-pixel targets with low error rates, it was generally time consuming to use. Furthermore it forced the user to break down a two-dimensional pointing task into two separate one-dimensional tasks, both mentally and physically. Mentally, the user needs to visualize how the cross point moves as a result of the handle movement, physically, one has to control one line (lever) at a time.

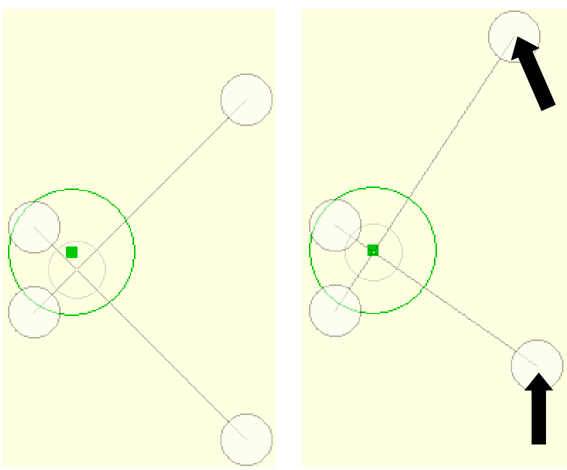

Figure 2. The *Cross-Lever* technique. The user deploys the *Cross*-Lever as close as possible to the highlighted target (left). To adjust the intersecting point to within the target the user drags the uppermost handle upwards and left and the lowermost handle upwards (right). The two other handles are not used and therefore do not move. The smaller circle surrounding the intersecting point is the activation area of the *Cross-Lever*.

Discrete Step Wise Solution – *Virtual Keys*

The second technique, *Virtual Keys*, is a small step from the existing practice of using physical cursor keys for precision control. Instead of using physical keys, *Virtual Keys* uses four graphical arrow keys and an activation key, all positioned on a side panel, to control the position of a crosshair cursor (Figure 3). A typical sequence would be first deploying the crosshair by touching approximately on the target, adjusting it using the arrow keys, then tapping the activation key. Although faster than both *Take-Off* and *Cross-Lever* for high precision pointing, our informal tests also indicated deficiencies of this technique, primarily due to the eye gaze and hand movement back and forth between the target area and the virtual keys on the side panel. This drawback would probably be less pronounced in a physical keys solution, where tactile feedback may complement the visual when operating the keys.

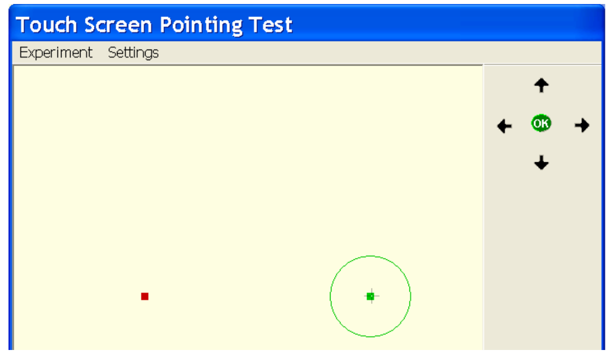

Figure 3. The *Virtual Keys* technique. Using the arrow keys the crosshair is adjusted into the green target.

Reducing Attention and Hand Travel – *Cross-Keys*

Our third design attempt, *Cross-Keys*, combined features from the first two. We moved the control keys in the *Virtual Keys* technique from the side panel to the crosshair, putting arrow keys at each of the four ends of the crosshair and the activation key at the centre of the crosshair (Figure 4). This reduced the need to move visual attention and the hand between the target area and control keys on the side panel. The first tap deploys the crosshair with the arrow keys, and if adjustments are needed one taps on the handles to move the crosshair in discrete steps. Once on target, the user taps the centre circle for activation. Depending on how much users missed the target with the initial touch they would either use the discrete step handle keys, or point again to get a better starting point. As in *Cross-Lever*, the graphic elements of *Cross-Keys* are all semi-transparent. An obvious limitation is if targets are situated very close to the screen edge, the handles would be pushed beyond the screen. Special design solutions for this problem include automatic panning, or displacement of handles. Our informal tests showed that the *Cross-Keys* technique was an improvement over *Virtual Keys* and overall worked well.

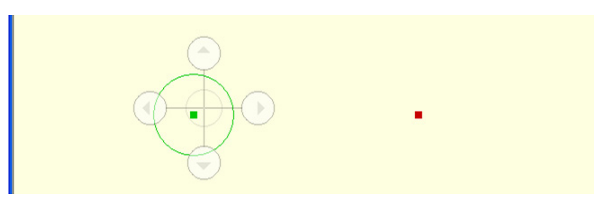

Figure 4. *Cross-Keys* – Shown in the picture are two targets and the *Cross-Keys* to be adjusted left and down to hit the current target.

Continuous and Integrated 2D Control – *2D Lever*

Cross-Keys achieves fine control with discrete key taps. Could a more fluid, continuous technique be designed to achieve the same goal? This led us back to the *Cross-Lever* technique, whose primary drawback was the need to break down a positioning task to two separate one-dimensional tasks. We hence designed our fourth new technique, *2D Lever* (Figure 5, 6a-b), consisting of a handle, a pivot point, and a tip with a crosshair to point to the target. The *2D Lever* is deployed by first touching as near the target as possible. Moving the handle causes the tip to rotate around the pivot with high precision, since the handle is much farther away from the rotation point than the tip. Moving the handle towards or away from the pivot point causes the tip to shrink or extend proportionally. When the tip of the *2D Lever* reaches the target, the user taps within the activation circle to select the target. Our informal tests showed that *2D Lever* was faster than *Cross-Lever* (Figure 2), but still did not reach the performance level of *Cross-Keys*.

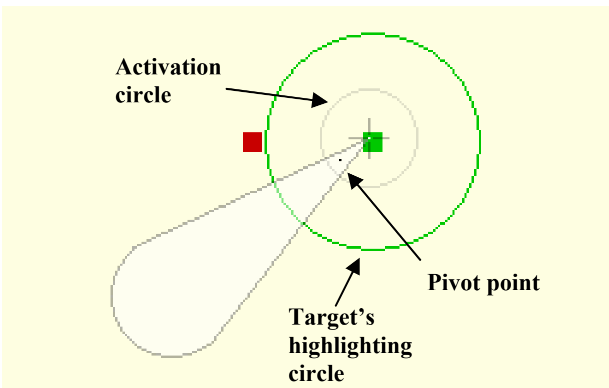

Figure 5. The *2D Lever* – the tip of the lever can be rotated or extended about the pivot (the small black point near the tip of the lever), with precision leverage.

Perfection by Simplification – *Precision-Handle*

Noting that one problem with the *2D Lever* was the inverted relation between the tip and the handle movement, we decided to skip real-world physics metaphors and remove the pivot point but maintain the amplified precision effect. The result is our fifth new design—*Precision-Handle*. As shown in Figure 6c-d, any movement made at the handle is also made at the tip but on a smaller scale, thus increasing precision. The handle will naturally stretch or shrink as the user manipulates it. To select the current crosshair position an activation circle around the tip was used as in previous techniques. By considering the screen area where the user initially taps, the layout of the handle can be optimized to provide large movement possibilities in all directions. If, for example, the target is on the lower left part of the screen the initial layout would be a handle pointing down and left, thus avoiding the problem of being placed outside the screen. In initial tests *Precision-Handle* was faster and better liked by users than *2D Lever*.

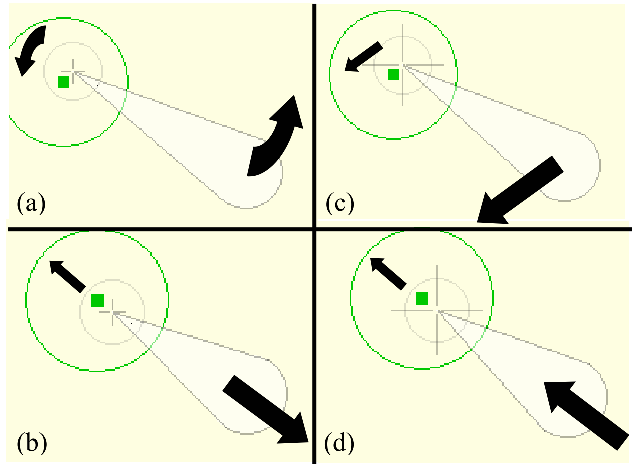

Figure 6. The *2D Lever* pivot point rotation (a) and translation (b) versus the *Precision-Handle* simplification (c, d).

Our iterative design exploration has produced five designs. Two deserved more rigorous evaluation—*Cross-Keys* and *Precision-Handle.* To measure their performance comparatively, we also studied two performance "anchors"—*Take-Off* and *Zoom-Pointing*, notwithstanding their limitations previously discussed. The basic goal of the current experiment was to evaluate the performance of these techniques at an elemental level which would be applicable to any task, and to provide results that may guide and be combined with studies with more specific context.

FORMAL EXPERIMENT

Twelve paid volunteers, eleven male and one female, with a mean age of 28.5 years (SD = 5.55 years), participated in the experiment. On a 1 to 5 scale, the participants' familiarity with touch screens averaged 2.33 (SD = .65), corresponding to "less than once per month". All of them used GUI computers on a daily basis. All but one of the participants were right-handed.

The main experimental apparatus was a commercial CRT-based 17″ touch screen, *Surface Acoustic Wave Touchscreen*, Model M17-SAW-S, made by Mass Multimedia, Inc. Its active display area was 320 x 240 mm and was set at 800 x 600 pixels resolution, with pixel size of 0.4 x 0.4 mm. Its refresh rate was set to 85 Hz. The screen was tilted to 40° to minimize fatigue [15].

A program was developed to present targets, provide the precision pointing techniques, measure user performance and log all necessary experimental data.

The experimental task was simple reciprocal target pointing. Two square targets of width *W*, separated by distance *D* were presented on the screen (See Figure 1 to 5). The *current* target, which alternates between the two, was colored green and surrounded by a green circle. Participants were instructed to select the green target as quickly and, more importantly, as accurately as possible.

Different audio tones were played for hit (inside the target) or error (out).

Since the focus of this study is precision pointing, target width *W* was set at 1, 2 or 8 pixels (0.4, 0.8, 3.2 mm) and distance *D* at 50, 250 or 500 pixels (20, 100, 200 mm). These size and distance combinations formed a wide range of index of difficulty (ID) values in Fitts' law terms, from 2.6 to 9.0 bits [5, 12].

For each of the 3 x 3 *W* and *D* combinations, whose order of appearance was randomly shuffled, 6 trials were performed. The first of the 6 trials was discarded since its distance was not controlled. Three blocks of all *W* and *D* combinations were repeated with each technique. Before the 3 blocks of actual data collection participants spent one practice block as a warm-up session. Each participant used all four techniques—*Zoom-Pointing*, *Take-Off*, *Cross-Keys*, and *Precision-Handle*. The precision gain of the *Precision-Handle* was fixed at seven times from the tip to the end of the handle. The order of these techniques was balanced by a Latin square pattern. A total of 4 (techniques) x 3 (blocks) x 3 (W) x 3 (D) x 6 (repetitions) test trials were collected from each participant.

The basic dependent measures were movement time (MT) and error rate (ER). Secondary measures concerning comfort, etc., were based on a questionnaire adapted from ISO 9241-9 [10]. Since we expected target size to have a critical impact on performance, participants were particularly asked to rate the different techniques according to "small" and "big" targets.

RESULTS

Repeated measure variance analyses were performed on both trial completion time and error rate, based on data collected from all three blocks of trials.

Learning Effect

Participants significantly improved their time performance over the three blocks of trials ($F_{2, 22} = 16.32$, $p < .0001$) but their error rate did not change significantly ($F_{2, 22} = 1.9$, $p = .17$). Since there was no significant interaction between trial blocks and interaction techniques, for either completion time (Technique x Block: $F_{6, 66} = 0.8$, $p = .5$) or error rate (Technique x Block: $F_{6, 66} = 0.672$, $p = .67$), we used data from all three blocks for the rest of the analysis.

On average (across all sizes and distances), for trial completion time *Zoom-Pointing* was significantly faster than all other techniques ($p < .0001$), and *Precision-Handle* was significantly faster than *Take-Off* ($p < .05$), as shown by post-hoc analyses, (Fisher's PLSD). For error rate, participants made significantly more errors with *Take-Off* than with any other techniques ($p < .0001$). The differences among the others were not significant ($p > .5$)

Size Impact

Target size had a significant impact on both completion time ($F_{2, 22} = 209.5$, $p < .0001$) and error rate ($F_{2, 22} = 22.74$, $p < .0001$). Target size difference also significantly affected the relative time performances of the four techniques, as shown by Technique x Size Interaction: $F_{6, 66} = 20.79$, $p < .0001$ (Figure 7).

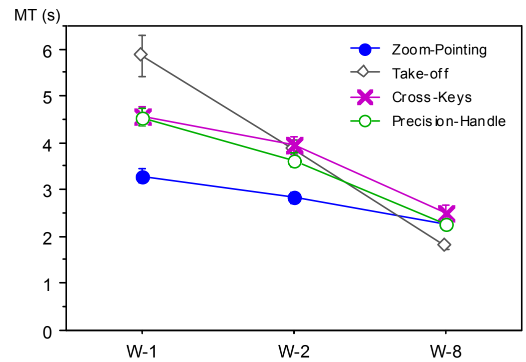

Figure 7. Mean completion time (s) as a function of target size, with 95% confidence error bars.

Particularly noteworthy is the performance of *Take-Off* in relation to target size. For small 1-pixel targets, *Take-Off* took longer than other techniques. For 2-pixel targets, *Take-Off* was comparable to *Cross-Keys*. For large targets (8-pixels), it took less time than any other techniques.

The relative error rate of the four techniques also depended on target size (Technique x Size Interaction $F_{6, 66} = 25.6$, $p < .0001$). In particular, for 1-pixel or 2-pixel targets, *Take-Off* was dramatically more error prone than the other techniques. For large target (8-pixels), *Take-Off* was comparable to other techniques. See Figure 8.

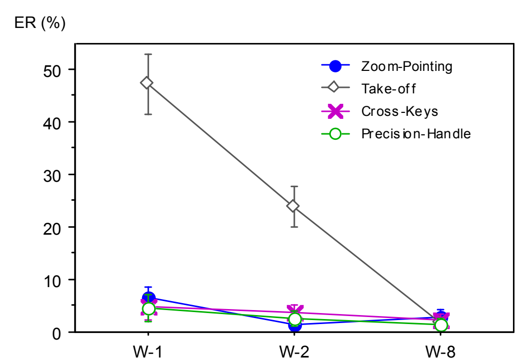

Figure 8. Mean error rate (%) as a function of target size with 95% confidence error bars.

Distance Impact

The distance between targets had a significant impact on completion time ($F_{2, 22} = 12.40$, $p = .0002$, Figure 9), but not on error rate ($F_{2, 22} = 0.61$, $p = .55$, Figure 10). Technique x Distance Interaction had borderline significance: $F_{6, 66} = 2.52$, $p = .03$, but $p = .083$ with Greenhouse-Geisser correction.

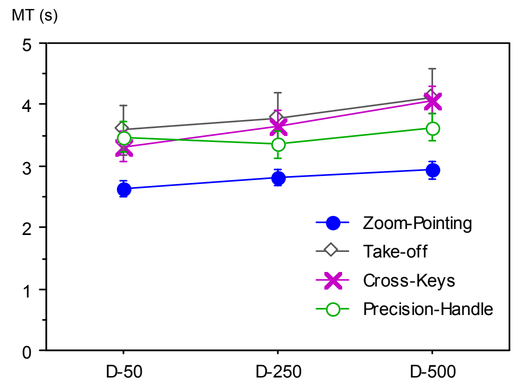

Figure 9. Distance x Technique Interaction plot for time in seconds, with 95% confidence error bars.

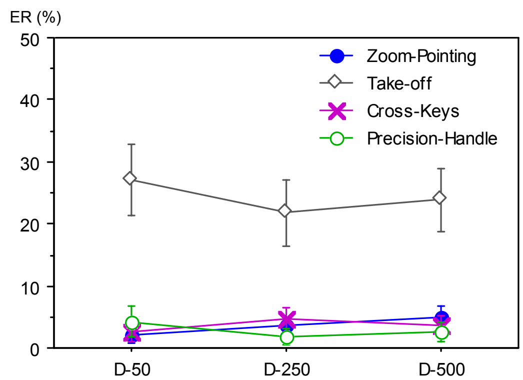

Figure 10. Distance x Technique Interaction plot for error rate (%), with 95% confidence error bars.

The magnitude of movement time difference caused by the 10-fold distance change in the experiment, however, was relatively small. This is plausible because all techniques studied, including *Take-Off* (first land and then drag the cursor), involved multiple stages of operation that makes distance less relevant. It is noticeable that, defying Fitts' law tendency, the mean completion time of *Precision-Handle* was even longer for the shortest than for the middle distance (Figure 9), although not significantly ($p = .99$).

Fitts' Law Analysis

Fitts' law [5], a human performance model for reaching tasks, has been widely used in studying pointing devices and techniques, since Card, English and Burr [4]. In 2000, Fitts' task was officially adopted by *ISO 9241-9* as one of the standard frameworks in which to study input devices [10]. Our current experiment was indeed designed in light of Fitts' law—both target distance and target size were manipulated.

It is questionable, however, if it is valid to parameterize our results with Fitts' law equation $MT = a + b \log_2(D/W + 1)$, where a and b are constants, and D and W are distance and width of the targets. Involving multiple steps, the interaction techniques in our study were more complex than the perceptual-motor mechanism involved in the typical single aiming movement in Fitts' law studies. Given that the D/W ratio determines MT according to Fitts' law, the impact of change in D or W should be the same accordingly. This was clearly not the case in our results.

If "force fitting" our data with a Fitts' law regression (Figure 11), we obtain a rather poor fit between the Fitts' law model and the actual data collected, with r^2 value at .71, .57, .74, .47 for *Zoom-Pointing*, *Take-Off*, *Cross-Keys*, and *Precision-Handle* respectively. These are much lower than the values of 0.95 or greater found in conventional one-step pointing tasks (e.g. [1]). In Figure 11 data points from left to right correspond to indexes of difficulty (*IDs*) originating from size 8, 2, 8, 1, 8, 2, 1(2) and 1 pixels respectively. Note that the 7th point was a combination of *D/W* 250/1 and 500/2, whose components are illustrated by open circles for the *Take-Off* plot. It is evident that target size (scale) rather than *ID* (stemmed from *D/W* ratio) was the more dominant determinant of completion time. To obtain a good fit, we would have to treat *ID's* resulting from different sizes separately, which would not summarize data of each technique with one equation.

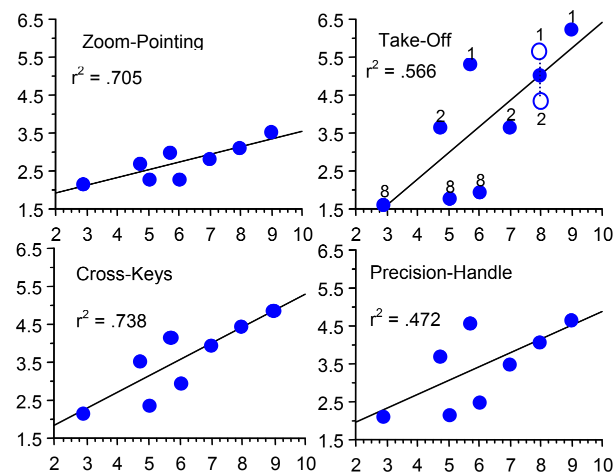

Figure 11. Fitts' law regression plots. Horizontal axis is Fitts' ID (bits) and vertical is trial completion time (s). For Take-Off data points are labeled with target size.

As recommended in [10], each input system can be potentially characterized by throughput (*TP*, or *index of performance*) based on Fitts' law regression, which ideally converts speed and accuracy into a single metric. We rejected such a temptation for two reasons. First, the actual performance did not match the Fitts' law model well. To use *TP*, a derivative concept from Fitts' law, to characterize the techniques is problematic. Second, it has been shown that using *TP* as a single dimensional metric of input performance is not logically valid [18].

However, the perspective brought by Fitts' law in terms of size and distance effects nonetheless provided a useful framework for the current study.

Subjective Evaluation

Participants rated the four techniques along the following dimensions: mental effort, accuracy, operation speed, fatigue concerning finger, hand, arm and eyes, general comfort, ease of use, and a concluding grade in relation to target size, on a 5 point scale (Table 1). Overall the four techniques received significantly different ratings: $F_{1, 33} = 33.5$, $p < .0001$. Post-hoc tests (Fisher's PLSD) show that *Take-Off* received a significantly lower score than all other techniques ($p < .0001$). *Zoom-Pointing* received significantly higher ratings than *Cross-Keys* ($p = .0018$), but not than *Precision-Handle* ($p = .14$).

Table 1. Mean subjective rating, from 1 (most negative) to 5 (most positive).

	Zoom-Pointing	*Take-Off*	*Cross-Keys*	*Precision-Handle*
mental	4.75	3.42	4.08	4.33
accuracy	4.50	1.67	4.00	4.08
speed	4.25	2.42	3.08	4.08
hand fatigue	3.33	2.83	3.83	3.75
eye fatigue	4.67	2.33	3.33	3.08
comfort	4.00	1.67	3.33	3.67
ease	4.33	3.08	4.08	4.17
small targets	4.58	1.08	3.92	4.00
large targets	3.58	4.75	3.92	4.50

If we total the scores in the 9 dimensions, with 45 as the maximum possible value and 9 as the minimum, *Zoom-Pointing* received a score of 38.0, followed by *Precision-Handle* 35.7, *Cross-Keys* 33.6, and lastly *Take-Off* 23.3. This result is congruent with completion time and error rate results.

As to individual dimensions, the only relatively low score for *Zoom-Pointing* was fatigue concerning arm and hand, probably due to movement back and forth between targets and the zoom button. As expected, fatigue concerning eyes was very low for zooming compared to the other techniques ($p < .0015$), since the user does not have to aim at a pixel level. Also worth mentioning was the fact that participants felt that operation speed for *Precision-Handle* was very close to *Zoom-Pointing* and both were felt considerably faster than the other two techniques ($p < .007$). Considering target size, *Zoom-Pointing* scores the best and *Take-Off* the worst for small sizes, whereas they change places for big targets. Second for both small and big targets was *Precision-Handle*, followed by *Cross-Keys*.

Comparing *Zoom-Pointing* and *Precision-Handle*, the ratings were significantly different in only two dimensions: eye-fatigue and big targets. For the former zooming is more positive ($p = .0002$), the later *Precision-Handle* more positive ($p = .0064$). Not far behind is *Cross-Keys* which received slightly worse grades.

CONCLUSIONS AND DISCUSSION

Our main goal in this work was to design techniques allowing users to precisely point at single pixels without resolving to zooming, which does not maintain the complete view of the entire area of interest. Our iterative design-test exploration yielded two promising techniques for such a purpose—*Cross-Keys* and *Precision-Handle*. We subjected these two techniques to a formal user study, together with two baseline techniques—*Zoom-Pointing* and *Take-Off*—to evaluate their fundamental performance characteristics. *Zoom-Pointing* performed well (faster speed with the same error rate) in this experimental task, especially in dealing with small targets. Once again zooming served as a performance anchor point in this study and its main drawback of losing overview can in many actual applications be problematic.

Take-Off, the best-known touch screen precision handling technique in the literature, fared poorly for small targets in comparison to the two new techniques. Special stabilization algorithms have been used to enhance the *Take-Off* technique in the past [16]. This experiment did not use any other software than the built-in drivers. However, the *Take-Off* performance in this study is comparable or superior to its original reported *stabilized* results in both time and error rate, notwithstanding experimental set-up differences [16]. This is probably due to the more modern touch screen hardware and drivers used in the current study. For 1 and 2 pixel targets, *Take-Off* error rate was much higher than *Precision-Handle* and *Cross-Keys* and for 1-pixel targets its speed was also much slower than the new techniques. This was also indicated in the subjective evaluation. For larger targets (8 pixels), however, *Take-Off* was faster than all other techniques (although not significantly except for *Cross-Keys*), with similar error rate, and rated highly in the subjective grading. *Take-Off*'s one-step nature makes it fast when the target is large enough, but hard to operate accurately when aiming at single pixels.

Cross-Keys allowed the users to select small targets with low error rate. The discrete movement appeared to make exact adjustments easy. On the negative side, to repeatedly tap on the handle, which moved with the crosshair, makes parallax and calibration problems more critical, particularly when the targets are on the outer sides of screen [11]. Slightly bigger handles may lessen this problem and also improve *Cross-Keys'* performance. Unstructured interviews also revealed that participants sometimes had difficulty seeing the crosshair and handles with the *Cross-Keys* technique because their finger and hand obscured them.

Precision-Handle performed with satisfactory speed and accuracy for both small and larger targets, thanks to the precision leverage effect. Subjectively, it was considered to be very close to *Zoom-Pointing* in most dimensions.

Some of these techniques can certainly be further modified without losing their basic characteristics. For example, it is possible to make *Precision-Handle* a one-step technique by using finger lifting, rather than a separate tap, as selection event.

Our exploration can be considered as designing "interaction instruments" [2] to overcome the limitations of bare hand direct manipulation. *Take-Off*, the prior art to our work, can also be considered as such because it offsets the user's finger action and the end-effecter (cursor). *Precision-Handle* can in fact be considered a further evolution of *Take-Off*—not only does it offset the position between finger action and end effect, but also the movement scales, therefore providing a precision leverage effect.

As demonstrated in the experimental results, these instruments are not universally superior or inferior to one another. In practice it is probably more desirable to switch tools according to different needs, just as in the physical world we use pliers, wrenches, screw drivers and other tools selectively. In the case of touch screens, when the targets are large enough, bare hand pointing is undoubtedly a very good choice. When the targets are smaller than a finger width but not at the pixel level, users may select *Take-Off* as their tool. For pixel level precision pointing, *Precision-Handle* shows promising attributes considering speed, accuracy and comfort. Discrete-tapping based *Cross-Keys* is likely to be very exact, suitable for the finest adjustments. When maintaining a complete view is not important, *Zoom-Pointing* can be the best choice. Well-designed user interfaces should provide a set of tools appropriate to its targeted application and an efficient and clear mechanism to support the selection and switching of these tools.

ACKNOWLEDGEMENTS

This work was partly done at Linköping University where SZ was a guest professor. We thank Ben Shneiderman for commenting on an early draft of this paper. We also thank Lars Eriksson and Alison Sue for their assistance.

REFERENCES

1. Accot, J. and S. Zhai. More than dotting the i's - foundations for crossing-based interfaces. *Proc. CHI* 2002. p. 73 - 80.
2. Beaudouin-Lafon, M. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. *Proc. CHI* 2000. p. 446-453.
3. Bederson, B.B. and J.D. Hollan. PAD++: Zooming graphical interface for exploring alternative interface physics. *Proc. UIST*. 1994. p. 17-26.
4. Card, S.K., W.K. English, and B.J. Burr, Evaluation of mouse, rate controlled isometric joystick, step keys and text keys for text selection on a CRT. *Ergonomics*, 1978. **21**: p. 601-613.
5. Fitts, P.M., The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 1954. **47**: p. 381-391.
6. Furnas, G.W. and B.B. Bederson. Space-scale diagrams: understanding multiscale interfaces. *Proc. CHI* 1995. p. 234-241.
7. Guiard, Y., M. Beaudouin-Lafon, and D. Mottet. Navigation as multiscale pointing: Extending fitts' model to very high precision tasks. *Proc. CHI* 1999. p. 450-457.
8. Guiard, Y., F. Gourgeouis, D. Mottet, and M. Beaudouin-Lafon. Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. *Proc. People and Computers XV: Joint Conference of IHM 2001 and HCI 2001*. 2001: Springer. p. 573-587.
9. Hornbæk, K., B. Bederson, and C. Plaisant, Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction*, in press.
10. ISO, *ISO 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices*, 2000, International Organization for Standardization.
11. Leahy, M. and D. Hix. Effect of touch screen target location on user accuracy. *Proc. The Human Factors and Ergonomics Society Annual Meeting*. 1990. p. 370-374.
12. MacKenzie, I.S., Fitts' law as a research and design tool in human computer interaction. *Human Computer Interaction*, 1992. **7**: p. 91-139.
13. Potter, R.L., L.J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI,* 1988. p. 27-32.
14. Ren, X. and S. Moriya, Improving selection performance on pen-based systems: A study of pen-input interaction for selection tasks. *ACM Transactions on Computer Human Interaction*, 2000. **7**(3): p. 384-416.
15. Schultz, K., D. Batten, and T. Sluchak, Optimal viewing angle for touch-screen displays: Is there such a thing? *International Journal of Industrial Ergonomics,* 1998. **22**: p. 343-350.
16. Sears, A. and B. Shneiderman, High Precision Touchscreens: Design Strategies and Comparison with a Mouse. *International Journal of Man-Machine Studies*, 1991. **43**(4): p. 593-613.
17. Shneiderman, B., Touch screens now offer compelling uses. *IEEE Software*, 1991. **8**(2): p. 93-94, 107.
18. Zhai, S., *On the Validity of Throughput as a Characteristic of Computer Input*, IBM Research Report RJ 10253, 2002, IBM Almaden Research Center: San Jose, California.