MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Absolute vs. Relative Direct Pen Input

Clifton Forlines, Daniel Vogel, Nicholas Kong, Ravin Balakrishnan

TR2006-066 June 2006

Abstract

We present the findings from two experiments designed to explore the effect of absolute vs. relative direct pen interaction on both small and large scale displays where the input and display spaces are co-incident. An absolute mapping - where there is a one-to-tone correspondence between the pen and cursor positions - was found to be superior to a relative mapping - where the pen and cursor positions can be offset with a variable mapping - for all distances on the small screen of a Tablet PC; however, on a large wall-sized display, the relative mapping outperformed the absolute mapping in situations requiring cursor movement over large distances. Our findings can inform the design of pen interfaces, in particular for large scale displays.

Absolute vs. Relative Direct Pen Input

Clifton Forlines[1], Daniel Vogel[2], Nicholas Kong[2], Ravin Balakrishnan[2]
[1]Mitsubishi Electric Research Labs
Cambridge, MA USA
forlines@merl.com
[2]Department of Computer Science
University of Toronto
dvogel | ravin@dgp.toronto.edu

Abstract

We present the findings from two experiments designed to explore the effect of absolute vs. relative direct pen interaction on both small and large scale displays where the input and display spaces are co-incident. An absolute mapping – where there is a one-to-one correspondence between the pen and cursor positions – was found to be superior to a relative mapping – where the pen and cursor positions can be offset with a variable mapping – for all distances on the small screen of a Tablet PC; however, on a large wall-sized display, the relative mapping outperformed the absolute mapping in situations requiring cursor movement over large distances. Our findings can inform the design of pen interfaces, in particular for large scale displays.

Key words: pen-based input, pen, relative input, absolute input, wall display, tablet computing, direct input

1 Introduction

The growing popularity of pen-based devices makes direct input – pointing and manipulating a cursor by touching the surface of a display directly – a viable alternative to indirect input such as using a mouse where there's a spatial separation between the input device and output display. When combined with direct manipulation graphical interfaces, direct input provides a strong affordance since users can simply touch the virtual entities they wish to work with [17].

Most pen-based devices use an absolute device-cursor mapping where the cursor is placed directly under the pen tip (Figure 1a,b). Although this may be the most obvious mapping for direct pen-based interaction, there are drawbacks. For example, the user's hand occludes part of the display and accurate selection is hindered by parallax error [6]. But perhaps the most limiting aspect of an absolute mapping is related to large display sizes. Until recently, most devices using direct input have been small enough that all areas of the display are within easy reach, and an absolute mapping works well in this case. However, as displays increase in size, an absolute mapping may become less desirable since users have to stretch their arm, twist their body, or, in the case of wall-sized displays, physically walk over to select a distant object. In some very large displays (e.g., very tall displays), it may even be impossible to directly touch all parts of the display.

Direct pen input does not have to use an absolute mapping between pen tip and cursor. Instead, a relative mapping can be employed, where the cursor and pen are no longer co-incident (Figure 1c,d). Thus, parallax error is no longer a concern, theoretically making the selection of small targets more accurate, and the hand may be positioned to avoid occluding important content. Also, with a control-display (CD) gain greater than one, relative input can move the cursor a great distance with only a short pen movement. Combined with a clutching mechanism, distant objects may be reached without the user adjusting their body position.

Intuitively, we hypothesize that an absolute mapping will perform well when the distances to be traversed are small, whereas a relative mapping might be best when distances get larger. However, the affordance of an absolute "under-the-pen" mapping may be so strong that users could find using a relative mapping difficult or unnatural, impacting performance even at large distances. Further, using a relative mapping for targets that are far away might result in the target being harder to see and thus select than in an absolute mapping where the user is always visually close to the target. To explore these issues, we conducted two experiments that compare performance between absolute and relative mappings for direct pen input in a canonical target selection task, on both small and large scale displays. The results provide guidelines on how we should design pen interfaces for displays of varying sizes.

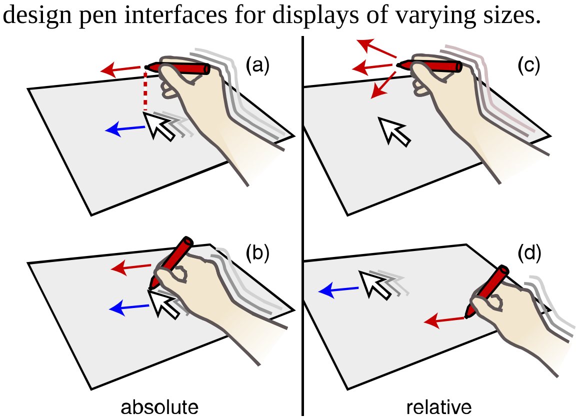

Figure 1. Absolute and relative direct pen interaction. (a) absolute hovering: the cursor tracks the pen tip; (b) absolute dragging: cursor directly under pen tip; (c) relative hovering: cursor does not track (a "clutching" operation); (d) relative dragging: pen controls movement of cursor from a distance.

2 Related Work

Graham and MacKenzie [7] compared selection performance using direct *physical* and indirect *virtual* touching. In the physical condition, users selected targets with their hand directly on a physical surface, but in the virtual condition, the user's hand was hidden and rendered as a "virtual finger" on a display. There was no performance difference between techniques for the initial movement phase, but virtual touching was slower in the second movement phase as the hand decelerated to select small 3 to 12 mm targets. This suggests that direct input can outperform indirect input.

Regarding the performance of absolute vs. relative mappings, researchers have arrived at different conclusions. Sears and Shneiderman [16] compared relative indirect mouse input to absolute direct touchscreen input. Their experiment used a 27.6 by 19.5 cm display with a mouse CD gain close to 1. They found that for targets 16 pixels in width and greater, absolute direct selection using the touchscreen was faster than relative indirect selection with a mouse. Further, for targets 32 pixels in width, absolute touchscreen selection resulted in about 66% fewer errors. Yet, even with the apparent superior performance for absolute direct touch input, participants still preferred mouse input. Meyer et al. [13] compared two absolute devices (touchscreen, indirect absolute pen) and three relative devices (mouse, trackball, mousepen – a relative indirect pen) on a desktop display. They found that when used in an indirect manner (with separated control and display space), the relative mousepen performed better than the absolute direct pen. In fact, they found all absolute input devices to be slower than the relative devices and concluded that "relative mapping is superior to absolute mapping."

These results are in contrast to that of Accot and Zhai [1] who found that for steering tasks users were about twice as fast with an 8"x6" indirect tablet in absolute mode than with a smaller indirect touchpad in relative mode.

3 Relative Mapping with a Direct Input Pen

Using a relative mapping with direct pen input is not a common interaction technique, so we now describe how pen movements are mapped to cursor movements to create relative direct interaction using the terminology of Buxton's 3-state model [3] (Figure 2). For an absolute pen, like that used with a Tablet PC, *State 0* input occurs when the pen is beyond the 1cm sensing range of the tablet resulting in no movement of the cursor; *State 1* input occurs when the pen is hovering within 1cm of the tablet's surface and the cursor tracks the location of the pen tip (Figure 1a); and State 2 input occurs when the pen is in contact with the tablet, allowing selection and dragging (Figure 1b).

For relative input, one needs to support not only tracking, dragging, and selection, but also clutching. The obvious method of moving the pen out of the hover range to clutch is undesirable since it requires the user to awkwardly lift their hand a certain distance from the display. Further, the distance at which the tablet stops sensing the pen differs between tablets, making it difficult for users to learn this threshold. Instead, when the pen is *lightly* in contact with the tablet, we make the cursor track (State 1 input) in accordance to movements of the pen. Lifting the pen even slightly away from the tablet surface signaled a clutching action and resulted in State 0 input (Figure 1c). Touching the pen on the tablet surface again returns to State 1 input, except that the cursor now moves relative to where it was before the clutching action took place (i.e., the cursor is not necessarily directly under the pen tip as in the absolute input situation). Pressing *firmly* with the pen resulted in State 2 selection and dragging (Figure 1d). The pressure sensing capabilities of most tablets make this pressure distinction trivially easy to implement.

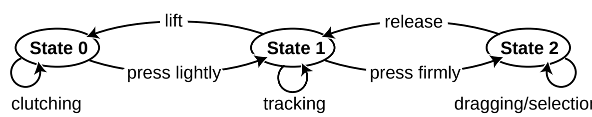

Figure 2. Buxton's 3-state interaction model [3], as applied to our relative pen input technique.

When tracking and dragging, we vary the CD gain between pen and cursor movement as a function of pen velocity – typically referred to as a *pointer acceleration function*. One advantage proposed by Jellinek and Card [8] is that this decreases the "footprint" of the input device suggesting that a large display area can be controlled from a small input area. We based our acceleration function on the one used in Windows XP [14], but altered the shape of its input/output velocity curve to provide more control at lower speeds and high gain factors at high speeds. We further tuned the scale of the function for each display size by applying scale factors independently to the velocity threshold and gain axes.

MacKenzie et al. [12] showed that using a pen in State 1 (tracking) results in better performance than in State 2 (dragging). Since our relative pen input method manipulates the cursor while usually in State 2 (dragging), all things being equal we should expect it to be slower than absolute input. However, relative input has a possible performance advantage. Balakrishnan and MacKenzie [2] show that pointing with the forearm is slower than the wrist which in turn is slower than the fingers. Even on a tablet sized display, we expect the reaching action of absolute pen input to require coordination between slower limbs, but with a properly tuned pointer acceleration function relative pen input can take advantage of the higher performance fingers and wrist.

4 Experiment One: Small Size Display

Our goal in this experiment is to compare absolute and relative direct pen input on the small display of a Tablet PC. Although all areas of the screen are easily within reach, users have to contend with issues like occlusion and fatigue when using an absolute mapping. As a baseline, we included mouse input as a relative indirect input device. Although Card et al. [4] found that mouse input was as fast and accurate as indirect pen input (with display and tablet occupying separate spaces), we wanted to see how it compares to direct pen input.

4.1 Participants, Apparatus, and Task

12 participants, 3 women and 9 men, 19 to 38 years old, were recruited from local universities and non-technical administration personnel in our lab. All were regular computer users, but did not have significant experience using pen input. Participants were paid $20 each, regardless of how they performed in the experiment.

The experiment was conducted on a 1.7 GHz *Toshiba Portege* M200 Tablet PC running Windows XP Tablet PC Edition, with a screen measuring 30.7 cm diagonally with a resolution of 1400x1050 pixels (56.7 pixels/cm). The display was configured in the tablet configuration (with its screen folded to cover the keyboard and track pad) and positioned horizontally on a desk in front of the user. Participants sat during the experiment and operated the tablet with a handheld pen in their dominant hand. During the mouse conditions, participants used a *Razer Viper* optical mouse with a 1000 counts/inch resolution. Our pointer acceleration function was tuned for this small display so that the cursor moves approximately 3:2 at low pen speeds and with some practice, higher speed movements can place the cursor at any location on the display without clutching.

A standard Fitts' [5][11] style 2D target acquisition task was used, which required participants to point and click on a series of targets positioned around the screen. The targets were drawn as green squares on a black background. When participants successfully clicked on a target, it would flash red and then another target would appear elsewhere on the screen. When a participant missed a target, an error sound was played.

4.2 Design

A repeated-measures within-subject factorial design was used. The independent variables were *input technique*: (absolute pen, relative pen, and mouse), *target distance* (306, 612, and 1225 pixels), and *target width* (4, 8, 16, and 32 pixels). The smallest width of 4 pixels was chosen because this is the smallest size target that Windows XP Tablet Edition requires users to be able to select. We originally planned to experiment with targets as small as 1 pixel; however, an informal evaluation showed that error rates were so high for these tiny targets that the selection time data was essentially useless. These tiny targets were also often confused with pieces of dust on the screen.

The order of presentation of the 3 *input techniques* was counterbalanced among the twelve participants. For each *input technique*, participants performed 6 blocks of trials after a set of 20 warm-up trials. Within each block, participants performed 13 selections for each of the 4 *target widths*. The first of these 13 selections was discarded, because of the uneven starting point of the pen at the start of each set. The remaining 12 selections included 4 repetitions of each of the 3 *target distances* presented in random order.

Participants could take breaks between trial sets for each *target width*, and were encouraged to take breaks between *input techniques*. Each participant performed trials for all *input techniques* in one session, which took about one hour. In summary, the design was as follows:

12 participants x
3 techniques (absolute pen, relative pen, mouse) x
6 blocks per technique x
4 target widths (4, 8, 16, and 32 pixels) x
3 target distances (306, 612, and 1225 pixels) x
4 repetitions
= 10368 selections in total.

Participants had to successfully click and release within the target before the next target would appear, even if this required multiple clicks. This effectively removes the possibility that participants may try to "race" through the experiment by clicking anywhere.

4.3 Results

Selection Time. Selection time was the time taken between the appearance of a target on screen and the first successful click on the target. Selection time data reported below do not include trials marked as errors (i.e., those where the first click occurred outside the target). 407 trials with hardware errors or selection times greater than three standard deviations from the participant's mean technique selection time were counted as outliers and removed (3.9% of our data). The large number of outliers was in part due to the smooth surface of the Tablet PC, which extended beyond the bounds of the display and the sensing hardware's range. Participants sometimes slid the pen into this "dead" area around the display during relative input trials, which caused the pointer to stop moving and led to longer selection times. Also, the pen used included a barrel button that participants would sometimes inadvertently press while moving to the target. Through examining the log files, we were able to remove trials during which this type of error occurred.

Since we recorded data for all 6 blocks, we expected to see a learning effect. A repeated-measures ANOVA showed that block had a significant effect on selection time ($F_{1,11}$ = 14.77, p = .003) with participants' performance improving with practice. There was a significant interaction between block and selection technique ($F_{1,11}$ = 7.20, p = .02), with participants improving faster with the relative pen input than either absolute pen or mouse input. Even with this interaction, the rank order of the three techniques did not change. Interestingly, block had no effect on selection error rate.

Repeated measures ANOVA showed that there was a significant main effect for *input technique* on selection time ($F_{1,11}$ = 77.13, p < .001), with a mean of 1.39s for absolute pen, 2.20s for relative pen, and 1.44s for mouse. A post-hoc pair-wise means comparison showed a significant difference between the relative pen and absolute pen (p < 0.001), between relative pen and the mouse (p < 0.001), but not between absolute pen and the mouse (p = 0.25).

As one would expect from Fitts' law, there was a significant main effect for both *target width* ($F_{1,11}$ = 488.25, p < .001) and *target distance* ($F_{1,11}$ = 227.40, p < .001) on selection time. There was a significant interaction between *input technique* and *target width* ($F_{1,11}$ = 12.85, p = .004) (Figure 3), and between *input technique* and *target distance* ($F_{1,11}$ = 17.44, p = .002) (Figure 4). There was also a significant *target width* and *target distance* interaction ($F_{1,11}$ = 17.70, p = .001).

No other significant interactions were observed relative to selection time.

Selection Error Rate. There was a significant main effect for *input technique* on selection error rate ($F_{1,11}$ = 8.7, p = .002), with a mean error rate of 15.6% for absolute pen, 9.9% for relative pen, and 15.5% for mouse. A post-hoc pair-wise means comparison showed a significant difference between relative and absolute (p=0.01) and between relative and mouse (p=0.001) but no significant difference between absolute and mouse (p=0.58). Unlike with selection time, there was no significant effect for block on selection error rate.

Not surprisingly, there was a significant effect for *target width* on selection error rate ($F_{1,11}$ = 36.20, p < .001) as well as a significant effect for *target distance* on selection error rate ($F_{1,11}$ = 4.98, p = .047). There was no significant interaction between *input technique* and *target distance* on error rate; however, there was a significant interaction between *input technique* and *target width* ($F_{1,11}$ = 10.78, p = .007) (Figure 5). Very small targets benefited greatly from relative input, with this benefit diminishing as target widths increased. No other significant interactions were observed relative to selection error rate.

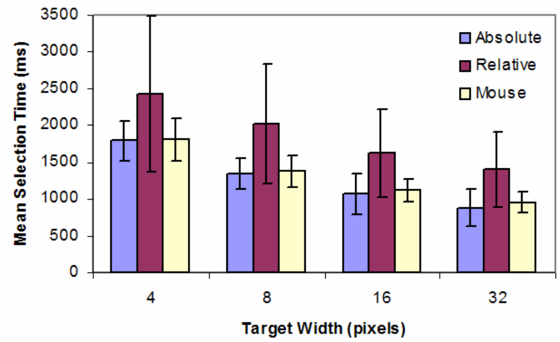

Figure 3. Mean selection times for each target width.

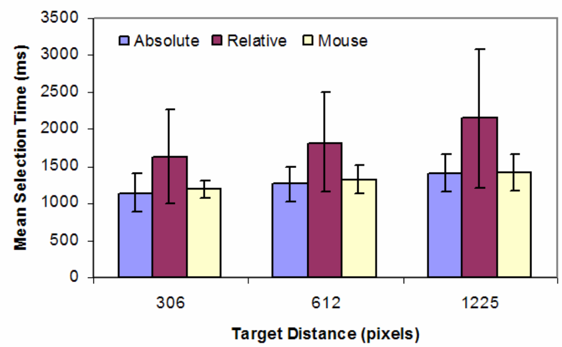

Figure 4. Mean selection times for each target distance.

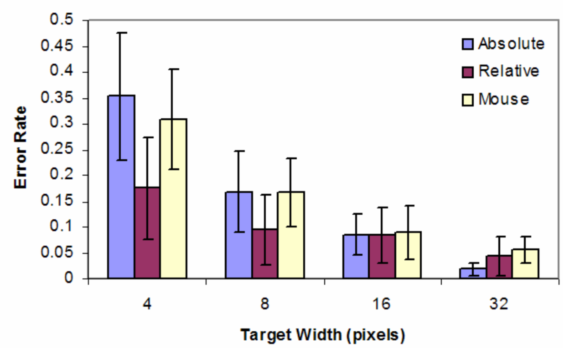

Figure 5. Mean error rates for each target width. Smaller targets were significantly easier to select with relative input.

Fitts' Law Analysis. The performance of a pointing technique can be modeled with Fitts' law [5][10]. The index of difficulty (*ID*) of a pointing task is a function of target distance (*D*) and width (*W*), and movement time (*MT*) can be predicted using:

$$MT = a + bID, \text{ where } ID = \log_2\left(\frac{D}{W} + 1\right)$$

where *a* and *b* are specific to a certain technique and are found using linear regression. The reciprocal of *b* is the index of performance (*IP*) which provides a measure of the technique's throughput. A higher *IP* indicates a more efficient technique. Table 1 summarizes the Fitts' law parameters for each of the three selection techniques. The high r^2 values indicate a close fit with the linear model for each technique.

Technique	Model	IP	r^2
Absolute	MT = -301 + 290 * ID	3.44	0.87
Relative	MT = -99 + 399 * ID	2.51	0.93
Mouse	MT = -42 + 255 * ID	3.91	0.88

Table 1. Fitts' law models for the three input techniques. All times are in milliseconds.

Subjective Feedback and Observations. At the end of each session, we asked participants to fill out a simple questionnaire and gave them a chance to tell us about their opinions of the three techniques. Participants strongly voiced their dislike of relative pen input in this setting. All participants correctly identified that relative input was the slowest of the three techniques. Three participants correctly identified that they were able to more accurately select small targets when using relative input than with absolute input, mentioning that they could always see the target in the relative condition. However, they still preferred absolute input when asked to pick between the two. Also, all but two participants listed relative input as the most tiring of the three techniques, likely because of the longer trial times.

While the difference was less pronounced, mouse input was rated slightly more favorably than absolute pen input in terms of speed, accuracy, and fatigue.

When using either pen input techniques, participants were forced to lean forward over the display more than when using the mouse. Because the display was positioned horizontally on the tabletop, this leaning may have provided a better view of the display, which could explain the superior performance of relative pen input over relative mouse input in terms of error rate.

5 Experiment Two: Large Size Display

In recent years, researchers have become increasingly interested in very large, wall-sized displays for use in collaborative settings (e.g., [9][15][18]). Many questions remain unanswered about the best input and interaction methods for working with these displays, and about how familiar desktop interaction techniques might best be changed (or scrapped) as they move to this new form factor. In our second experiment, we explore the relative merits of absolute and relative pen based selection on very large, wall-sized displays. At first glance, it might seem obvious that relative input would outperform absolute input for distant target selection; however, a deeper understanding of the complexities introduced by relative input when standing at a wall-sized display makes this outcome less obvious. When standing within arms' reach of a wall-sized display, distant targets become very hard to see. Further, relative input requires constant contact between the pen and display, which could lead to fatigue that negatively effects performance. We did not include the mouse in this experiment as it is not really a feasible input device for up-close interaction with wall size displays while standing.

5.1 Participants, Apparatus, and Task

We recruited 12 participants for this second study, 8 male and 4 female ranging in age from 17 to 37 years. These participants were students and professionals and were not paid for their participation.

We used a 5m wide, 1.8m high, back projected solid glass screen display (Figure 6), with imagery generated by 18 LCD projectors (each 1024x768 pixel resolution) in a 6x3 tiling. The effective resolution of this display is approximately 4730x1730 pixels (9.46 pixels/cm) because the projectors are overlapped to eliminate seams. A cluster of 18 PCs drive the projectors, with Chromium providing distributed graphics rendering (chromium.sourceforge.net).

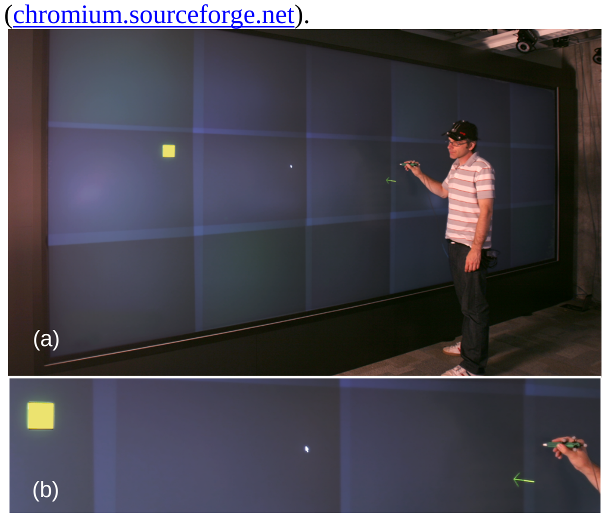

Figure 6. (a) using relative pen input on a large wall-sized display. (b) detail showing experiment stimuli: from left to right: target, cursor, direction arrow, pen input.

We used a Vicon (www.vicon.com) motion tracking system which streams sub-mm 3D coordinates of the pen tip at up to 120Hz. To reduce jitter in the pen position data, we use a dynamic recursive low pass filter which works without introducing lag during ballistic movements [19]. Our custom pen is instrumented so that pressing the tip hard against a surface activates a micro-switch to trigger click events. We also tracked the position and orientation of the participant's head using a Vicon-enabled hat. Our software was written in C++/OpenGL and is ran at 60 frames per second.

We tuned our pointer acceleration function for the large display such that the cursor moved approximately 1:1 at low pen speeds and with higher speed ballistic movements the cursor can be placed at any location on the display in a controlled manner without clutching.

A 2D target acquisition task similar to experiment one was used. To minimize visual search time, an arrow was displayed near the location of the previously selected target pointing in the direction of the next target.

5.2 Design

A repeated-measures within-subject factorial design was used. Our independent variables were: *input technique* (absolute and relative), *target width* (8, 16, 32, and 64 pixels), and *target distance* (946, 1892, 2838, and 3784 pixels). We chose to use more target distances in this study to reasonably cover the range of distances possible on the wall-sized display.

For each *input technique*, participants performed 7 blocks of trials. Within each block, 13 selections were made for each of the 4 *target widths*. The first of these 13 selections was discarded, because of the uneven starting point of the pen at the start of each set. The remaining 12 selections included 3 repetitions for each of the 4 *target distances* presented in random order.

Participants could take breaks between trial sets for each *target width*, and were encouraged to take breaks between *input techniques*. Each participant performed trials for both *input techniques* in one session, which took about one hour. In summary, the design was:

12 participants x
2 techniques (absolute, relative) x
7 blocks per technique x
4 target widths (8, 16, 32, 64 pixels) x
4 target distances (946, 1892, 2838, 3784 pixels) x
3 repetitions
= 8064 selections in total.

As in the first experiment, participants had to successfully click and release within the target before the next target would appear so participants couldn't "race" through the experiment.

5.3 Results

Selection Time. Selection time was the time taken between the appearance of a target on screen and the first click on the target. Selection time data reported below do not include trials marked as errors. 67 trials with selection times greater than three standard deviations from the participant's mean technique selection time were counted as outliers and removed (1.2% of data).

Since we recorded data for all 7 blocks, we expected to see a learning effect. A repeated-measures ANOVA showed that after we removed the first 2 blocks, block did not have a significant main effect on selection time ($F_{1,11} = 3.83$, $p = 0.08$). Thus, we considered the first 2 blocks as practice trials, and used only the last 5 blocks in the rest of our analysis.

Input technique did not have a significant main effect in terms of selection time, with mean selection times of 3.09s and 3.03s for absolute and relative input respectively.

As one would expect, both *target width* and *target distance* had a significant main effect on selection time ($F_{1,11} = 384.82$, $p < 0.001$ and $F_{1,11} = 632.52$, $p < 0.001$ for *target width* and *target distance* respectively).

Most interestingly, there was a significant interaction between *input technique* and *target distance* ($F_{1,11} = 108.36$, $p < 0.001$). For short distances, the absolute input technique led to shorter selection times while for long distances, the relative input technique led to shorter selection times. Figure 7 shows the mean selection time for each *input technique* / *target distance* combination. No other significant interactions were observed for selection time.

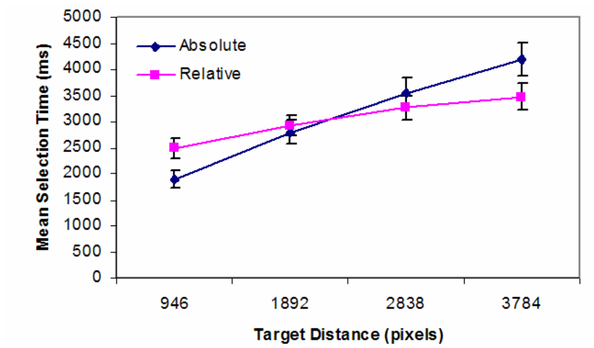

Figure 7. Mean selection times for each target distance and input technique combination. The relative performance in terms of selection time between absolute and relative input switches order between a distance of 1892 and 2838 pixels.

Selection Error Rate. A repeated-measures ANOVA showed a significant main effect for *input technique* on selection error rate ($F_{1,11} = 9.93$, $p = 0.009$). Participants averaged a 2.6% error rate when using absolute input, and a 5.4% error rate when using relative input.

Not surprisingly, *target width* had a significant main effect on selection error rate ($F_{1,11} = 26.55$, $p < 0.001$), with smaller targets having higher error rates (Figure 8). On the other hand, *target distance* had no significant effect on selection error rate (Figure 9). No significant interactions were observed relative to error rate.

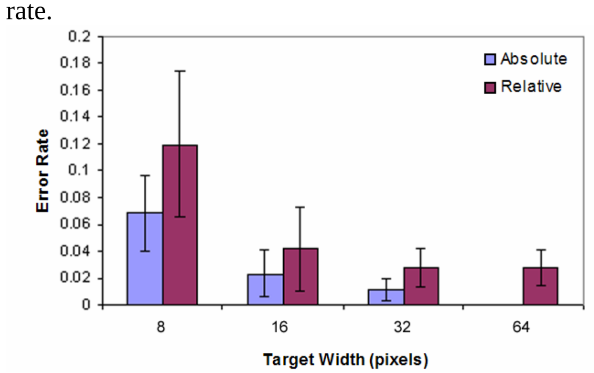

Figure 8. Mean error rates for each target width.

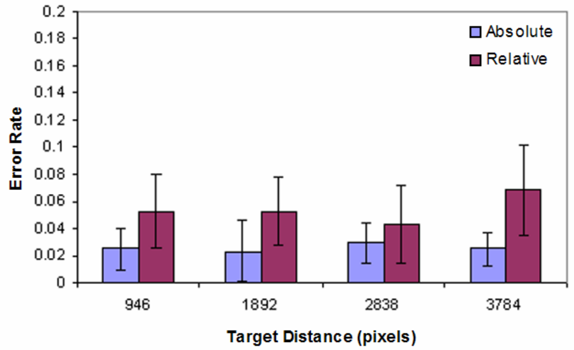

Figure 9. Mean error rates for each target distance.

Fitts' Law Analysis. Table 2 shows the Fitts' law model for each selection technique. It is interesting that the absolute technique has a poorer fit to the linear model. This is likely due to the different combination of wrist, hand, arm and leg movements that a participant had to use for absolute pointing as opposed to mainly wrist/arm movements for relative pointing. Given that the wrist, hand, arm and legs have different performance characteristics, it is not surprising that their combined use does not fit a single linear model with a high correlation. Unfortunately, we did not track the various limbs separately to perform a multi-model analysis. This single model analysis nonetheless provides some sense of the relative performance of the two techniques.

Technique	Model	IP	r^2
Absolute	MT = -847 + 599 * ID	1.67	0.72
Relative	MT = 383 + 402 * ID	2.49	0.96

Table 2. Fitts' law models for the two input techniques. All times are in milliseconds.

Visual Search. With a large wall-size display, users could potentially take considerably different amounts of time to visually locate the target during each trial depending on how far away the target is from the user's current position. Since most eye-trackers do not work effectively at this scale, we approximated gaze by tracking the position and orientation of participant's head and use this information to provide a rough estimate of the time taken to visually locate the target. Specifically, we recorded the amount of time between the start of a trial and the time when the angle between the head orientation vector and the vector connecting the participant's head and the target fell below a threshold.

There was little difference between the mean search times for the two input techniques (0.91s and 0.86s for absolute and relative input respectively). As one would expect, *target width* had a significant main effect on search time ($F_{1,11} = 7.44$, $p = 0.02$), with smaller targets taking more time to find. Interestingly, there was a significant interaction between *input technique* and *target distance* in terms of search time ($F_{1,11} = 26.41$, $p < 0.001$). With absolute input, *target distance* was a good predictor of search time, with distant targets taking longer to find; however, with relative input, target distance was less useful in predicting search time (Figure 11). This finding is probably explained by the strategies employed by participants. With absolute input, participants had to walk to and stand in front of targets to select them, so *target distance* was a good measurement of the true distance between a participant's eyes and the target. With relative input, participants tended to stand at the center of the display, minimizing body movement while maximizing visibility of the wall. In this case, target distance did not necessarily provide a true measure for the distance between participant and target. No other interactions were observed relative to search time.

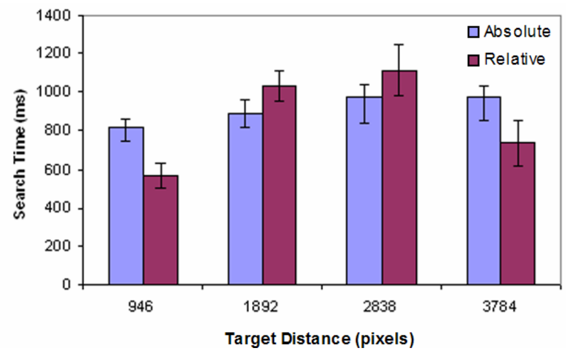

Figure 11. Search time for each target distance.

The difference in technique selection times could also be partly due to differing strategies for using the direction arrows. In the relative technique, participants looked in the direction of the arrow first, visually located the target, and then began manipulating the pen. In the absolute technique, participants would immediately start walking in the direction of the arrow and visually acquire the target on the way. Figure 12 shows search time as a percentage of selection time. That searching was a larger percentage of selection time for relative than for absolute input for distant targets, lends weight to our hypothesis that the visual target acquisition strategy used in the relative technique may have added as much as half a second to selection times.

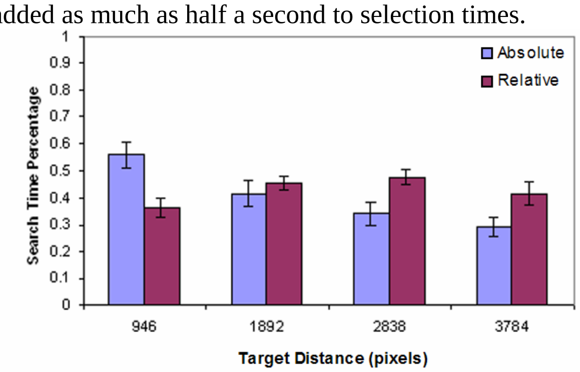

Figure 12. Search time as a percentage of selection time for each target distance.

Subjective Feedback and Observations. As in the first experiment, we asked participants to fill out a simple questionnaire evaluating the two techniques in terms of perceived speed, accuracy, fatigue, and ease-of-use. In contrast to experiment one, 7 participants felt that relative was faster and only 3 picked absolute (others were undecided). They commented that once comfortable with the relative technique; it seemed faster since it eliminated having to walk. However, the majority said that the absolute technique was the least tiring, most accurate, and not surprisingly easiest to use.

Arm fatigue was an issue with the relative technique. After the first 1 or 2 blocks of trials of the relative technique, participants minimized clutching as a strategy to decrease selection time. A side effect was that participants tended to hold their arm out with the pen tip constantly held against the display surface during each set of trials. So it is not surprising that almost all participants said it was very tiring to hold their arm up in the relative technique. But somewhat surprisingly, more than half of the participants felt this was *more* tiring than having to walk back and forth in front of the display in the absolute technique.

6 Conclusion

We have presented the results of two experiments designed to compare absolute and relative pen input on two different display sizes. While absolute input was superior in terms of selection time for all target distances on a Tablet PC, relative input overtook absolute input for distant target selection on a wall-sized display.

The crossover point at which relative input performed better than absolute input occurred at a distance of around 2200 pixels (about 200cm) indicating that relative input is preferable for displays whiteboard sized or larger. Clearly, the next experiment in this line of research should focus on the range between 1892 and 2838 pixels to more closely narrow down the true distance at which the crossover between the performance of relative and absolute input occurs.

In terms of selection errors, relative input resulted in fewer selection errors on a small display, perhaps because of the lack of occlusion by the hand and pen and the lack of parallax between the system pointer and pen tip. On the wall-sized display, absolute input led to lower error rates, probably because targets were easier to see once the participants had moved to select them.

These findings can directly aid the designers of tablet-sized and wall-sized interfaces and give weight to the argument that pen based interaction techniques designed and tested on small devices should be scrutinized before they are ported to very large displays.

References

[1] Accot, J. & Zhai, S. (1999). Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. *CHI*. p. 466-472.

[2] Balakrishnan, R. & MacKenzie, I. S. (1997). Performance differences in the fingers, wrist, and forearm in computer input control. *CHI*. p. 303-310.

[3] Buxton, W. (1990). A Three-State Model of Graphical Input, *INTERACT*. p. 449-456.

[4] Card, S., English, W. and Burr, B. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 21. p 601-613.

[5] Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. of Exp. Psychology*, 47. p. 381-391.

[6] Goldberg, D. & Goodisman, A. (1991). Stylus user interfaces for manipulating text. *UIST*. p. 127-135.

[7] Graham, E. & MacKenzie, C. (1996). Physical versus virtual pointing. *CHI*. p. 292-299.

[8] Jellinek, H. & Card, S. (1990). Powermice and user performance. *CHI*. p. 213-220.

[9] Johanson, B., Hutchins, G., Winograd, T. & Stone, M. (2002). PointRight: experience with flexible input redirection in interactive workspaces.*UIST*. p. 227-234.

[10] MacKenzie, I. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7. 91-139.

[11] MacKenzie, I. & Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *CHI*. p. 219-226.

[12] MacKenzie, I., Sellen, A. & Buxton, W. (1991). A comparison of input devices in element pointing and dragging tasks. *CHI*. p. 161-166.

[13] Meyer, S., Cohen, O. & Nilsen, E. (1994). Device comparisons for goal-directed drawing tasks. *CHI extended abstracts*. p. 251-252.

[14] Microsoft. (2005). Pointer ballistics for Windows XP. Accessed on 14 Feb 2005, *www.microsoft.com /whdc/device/input/pointer-bal.mspx*.

[15] Scott, E., Richard, B., Rich, G., David, G., Frank, H., William, J., David, L., Kim, M., Elin, P., Ken, P., John, T. & Brent, W. (1992). Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. *CHI*. p. 599-607.

[16] Sears, A. & Shneiderman, B. (1991). High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. *International Journal of Man-Machine Studies*, 34(4). p. 593-613.

[17] Shneiderman, B. (1991). Touchscreens now offer compelling uses. *IEEE Software*, 8(2), p. 93-94.

[18] Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., & Steinmetz, R. (1999). i-LAND: an interactive landscape for creativity and innovation. *CHI*. p. 120-127.

[19] Vogel, D. & Balakrishnan, R. (2005). Distant freehand pointing and clicking on very large high resolution displays. *UIST*. p. 33-42.