HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device

Clifton Forlines, Daniel Vogel, Ravin Balakrishnan

TR2006-090 October 2006

Abstract

We present HybridPointing, a technique that lets users easily switch between absolute and relative pointing with a direct input device such as a pen. Our design includes a new graphical element, the Trailing Widget, which remains "close at hand" but does not interfere with normal cursor operation. The use of visual feedback to aid the user's understanding of input state is discussed, and several novel visual aids are presented. An experiment conducted on a large, wall-sized display validates the benefits of HybridPointing under certain conditions. We also discuss other situations in which HybridPointing may be useful. Finally, we present an extension to our technique that allows for switching between absolute and relative input in the middle of a single drag-operation.

ACM Symposium on User Interface Software and Technology (UIST)

HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device

Clifton Forlines[1], *Daniel Vogel*[2], *Ravin Balakrishnan*[2]

[1]Mitsubishi Electric Research Labs
Cambridge, MA USA
forlines@merl.com
www.merl.com

[2]Department of Computer Science
University of Toronto
dvogel | ravin@dgp.toronto.edu
www.dgp.toronto.edu

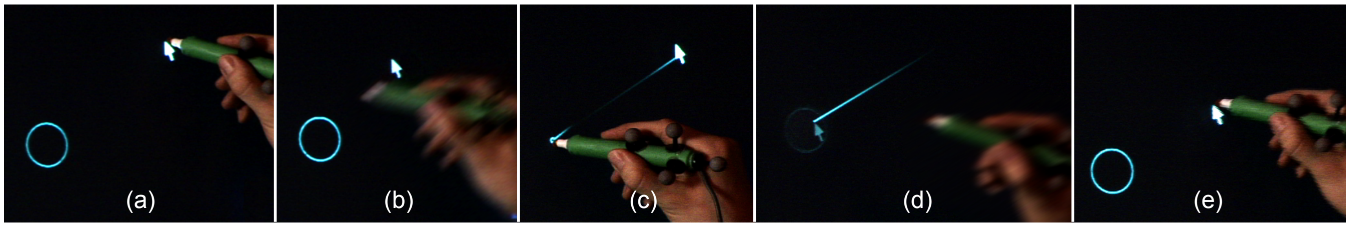

Figure 1. HybridPointing allows the user to switch between working in absolute or relative input.modes (a) By default, the pen works in absolute input mode, (b) a quick click on the circular *trailing widget* switches to (c) relative mode. (d) Lifting the pen a certain distance away from the display while in relative mode switches back to (e) absolute mode.

ABSTRACT

We present HybridPointing, a technique that lets users easily switch between absolute and relative pointing with a direct input device such as a pen. Our design includes a new graphical element, the *Trailing Widget*, which remains "close at hand" but does not interfere with normal cursor operation. The use of visual feedback to aid the user's understanding of input state is discussed, and several novel visual aids are presented. An experiment conducted on a large, wall-sized display validates the benefits of HybridPointing under certain conditions. We also discuss other situations in which HybridPointing may be useful. Finally, we present an extension to our technique that allows for switching between absolute and relative input in the middle of a single drag-operation.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: pointing, absolute input, relative input, direct-input, wall-sized display, multiple displays.

INTRODUCTION

When using a touch-sensitive or pen-based input device, manipulating graphical objects by directly touching the display provides a strong affordance for interaction. Direct manipulation occurs directly under a finger or pen-tip and users simply touch the graphical objects they wish to work with [26]. Direct input is growing in popularity as a viable alternative to indirect input, such as using a mouse, during which there is a spatial separation between the input device and output display.

Most direct input touch-sensitive or pen-based devices use an absolute device-pointer mapping in which the system pointer is positioned directly under a finger or stylus. While this is the most obvious, and arguably the most natural mapping, there are drawbacks. Hands, arms, fingers, and stylus all occlude portions of the display, an especially troubling issue for multi-user systems. For many rear-projection devices, accurate pointing and selection are hindered by parallax error [12]. For front-projection, using absolute input casts a shadow over the object of focus. While on a small display all areas of the interface are easily within reach, on a large display this may not be the case. Using an absolute mapping loses its desirability the more a user must stretch their arms, twist their waist, or physically walk to distant parts of a display. In extreme cases, it may become impossible to reach the extents of very tall displays. These difficulties only increase when working in a large, multi-display environment in which distant objects may not only be hard to reach, but also may require the user to interact across bevels or even gaps between displays.

Relative input overcomes many of these limitations, albeit perhaps at the cost of naturalness. The main benefit in terms of large, wall-sized displays is that distant targets can be manipulated without walking, as small movements of the input device can be mapped to large movements of the pointer. The opposite is true as well – relative input allows for more control over pointing as control-display (CD) gain ratios can be less than 1:1 for slow movements [17]. Target occlusion and parallax are less of a problem, dragging between displays is easily supported, and all areas of a large display are within easy reach. Finally, in a multi-user setting, users can reposition themselves such that they do not block the view of collaborators.

Normally system designers must choose between absolute direct input and relative indirect input when building their system; however, an input mechanism that supports fluid switching between relative and absolute mappings while using a direct input device might enable users to benefit from the best of both worlds. In this paper, we present such a technique, called HybridPointing, that let's users dynamically control the mapping between the movements of their direct input device and the system pointer.

RELATED WORK

Comparisons between Absolute and Relative Input

Regarding the performance of absolute vs. relative mappings, researchers have arrived at different conclusions. Sears and Shneiderman [24] 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 of absolute direct touch input, participants still preferred mouse input. Meyer et al. [18] 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."

On the other hand, Graham and MacKenzie [13] 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 in some situations.

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.

An example input device that supports absolute input in both a direct and indirect manner was presented by Parker et al. [21]. Their "Tractor-beam" input device is a stylus tracked in 3D and used like a laser pointer, with the system cursor positioned at the intersection of the virtual laser and the table. Selections are made by pressing a barrel-button on the stylus. When held against the display surface, the device acts as an absolute direct input device; however, when lifted from the display, interaction occurs at the position of the virtual "laser-dot". Through lifting and returning the pen tip to the display, the user switches between a direct and indirect absolute input device; however, accurate selection of distant targets with a laser pointer is well known to be difficult and error prone [19] making the usability of this input device questionable for large displays.

Multi-user Considerations

Several recent publications have addressed the issue of reach on large displays or across multiple displays [4, 5, 23]. Many of these solutions repurpose large areas of the workspace or display graphical feedback over a wide area of the display. While this approach is fine for an individual user on a wall-sized display, the heavy use of graphics may be inappropriate for multi-user workspaces in which other users may become distracted by these techniques.

Baudisch et al. [4] presented Drag-and-Pop as a means of moving a selected object to a distant target. When the technique is invoked, proxies for distant targets are drawn near the user where they are easily within reach. Reference lines connect these proxies to their true targets, and there is a good chance that these reference lines might cut through another's workspace in a multi-user setting. Similarly, Bezerianos et al. [5] presented the Vacuum technique for selecting distant objects. The Vacuum displays a large adjustable area of effect that can easily cover much of the display as distant targets are drawn close to the user. Like Drag-and-Pop, the disruption of other users working in the same space may reduce the benefits of addressing the reachability problem of large displays.

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 [7]. With an absolute pen, *State 0* input occurs when the pen is beyond the sensing range of the input device resulting in no movement of the pointer; *State 1* input occurs when the pen is hovering near the display surface and the pointer tracks the location of the pen tip; and *State 2* input occurs when the pen is in contact with the display, allowing selection and dragging, as shown in Figure 2.

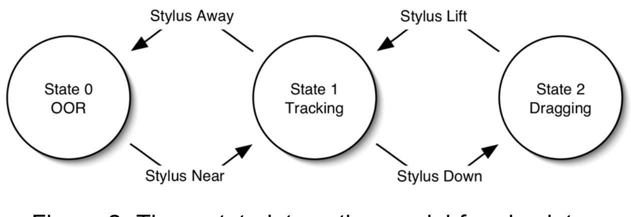

Figure 2: Three-state interaction model for absolute graphical input [7].

For relative input, one needs to support not only tracking, and dragging/selection, but also clutching. Figure 3 shows a modified three-state model for relative pen input. When the pen is *lightly* in contact with the display, we make the cursor track in accordance to movements of the pen. Lifting the pen even slightly away from the display surface signals a clutching action. Returning the pen to the display surface again returns to tracking, except that the cursor now moves relative to where it was before the clutching action took place (i.e., the cursor is no longer necessarily directly under the pen tip as in the absolute input situation). Pressing *firmly* with the pen results in selection and dragging. The pressure sensing capabilities of most tablets make this pressure distinction trivially easy to implement.

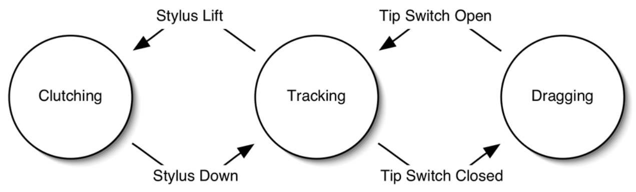

Figure 3: Three-state interaction model, as applied to relative pen input.

When tracking and dragging, we vary the CD gain between pen and pointer movement as a function of pen velocity – typically referred to as a *pointer acceleration function*. We based our acceleration function on the one used in Windows XP [17], 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.

INITIAL EXPERIMENTS

Intuitively, we hypothesized that an absolute mapping will perform well when target distances are small, whereas a relative mapping is 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, lowering 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 select than in an absolute mapping where the user is always visually close to the target.

To explore these issues, we conducted an experiment [11] in which we compared performance in a target selection task between absolute and relative mappings for direct pen input on a large wall-sized display. Figure 4 illustrates performance of both relative and absolute input for different target distances. Target distances in this study ranged from about 1000 to almost 4000 pixels, which corresponded to physical distances between 1m and 4m. As one can see from the chart, participants performed better when using absolute input for close targets, and performed better when using relative input for distant targets. This crossover in performance indicates that when working on a single large display that is over 2m wide, a user may benefit from being able to select an absolute or relative mapping when working with differently distanced targets.

We completed a similar study [11] in which participants used both absolute and relative pen input on a TabletPC with a 12.1" diagonal screen. Participants were significantly faster at and greatly preferred selecting targets with absolute input for all targets on this small display; however, for small targets, participants were more accurate when using relative input. This tradeoff among speed, preference, and accuracy indicates that users may benefit from being able to switch to a relative mapping when a high level of accuracy is required. Furthermore, this type of small personal device is increasingly used within a connected multi-display environment [15,23,25,27]. Allowing a user to perform input with an absolute mapping when working on their personal device while allowing them to fluidly switch to a relative mapping for controlling pointers on additional displays would be desirable.

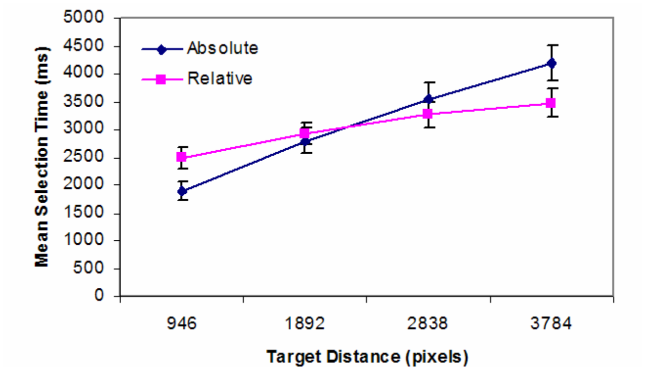

Figure 4: Selection times for targets at different distances for both absolute / direct input and relative / indirect input (from [11]). The crossover in performance between absolute and relative input occurred at distances of around 2m.

TRAILING WIDGETS

The accessibility problems of distant interface elements on a large display may be partially mediated for commonly used elements through a new class of widgets. We propose *trailing widgets* as a new way to keep user interface widgets like buttons, menus, palettes and handles "close at hand" with direct input devices. When using a mouse, context sensitive pop-up menus invoked with a right-click serve a similar purpose, but with a direct pen based interface, right-clicking is difficult and unnatural. But even if right-clicking is not a problem, researchers have argued for usability and performance benefits when widgets like palettes are located near the cursor and visible at all times [6]. Recent pen-specific research projects such as tracking menus [10] and hover widgets [14] address the need to keep commands near the pen, but tracking menus obscure the immediate area around the cursor and hover widgets rely on memorizing a set of pen gestures.

A trailing widget is visible at all times and does not obscure the immediate area around the cursor and, unlike magic lenses, does not need to be actively managed. The widget has a preferred position relative to the pen, which in our case is about 20cm to the South-West. The movement dynamics are tuned such that the widget stays out of the way during typical pen motion, but with a quick and deliberate pen movement, the widget can be selected before it moves away. We use a simple interpolated low pass filter to adjust the widget's movement dynamics based on the speed of the pen and its distance from the trailing widget. This makes the widget appear to float nearby while avoiding moving directly under the pen tip. The ballistic, direct pointing target selection needed to select the trailing widget is only reasonable with a pen-style input device; it is much more difficult with a mouse.

The HybridPointing technique described in this paper uses a very simple single trailing widget which was found to be effective, but it remains to be seen how far the concept of trailing widgets can be expanded. We plan to explore the design space of trailing widgets and to compare trailing widgets to similar tools in future work.

A HYBRID OF ABSOLUTE AND RELATIVE INPUT

The goal of HybridPointing is to enable the user to easily switch between absolute or relative input, using whichever input mode is most appropriate for the task at hand. A lightweight means of switching between the two mutually exclusive input modes is critical to the success of the technique. We initially explored pen gestures which "cast" the cursor out to switch to relative input and then "reeled" the cursor back in to switch to absolute. However, issues with reliable gesture recognition and the non-self revealing nature of gestures in general made this switching technique awkward and slow. Instead we designed a very simple switching method using two simple pen movements. Relative input mode is selected with a quick pen movement to trap a trailing widget -- a viscous target which follows the pen from a safe distance. Switching back to absolute mode is accomplished by simply lifting the pen past a hover threshold. Figure 5 illustrates the combination of the two three state input models for absolute and relative input.

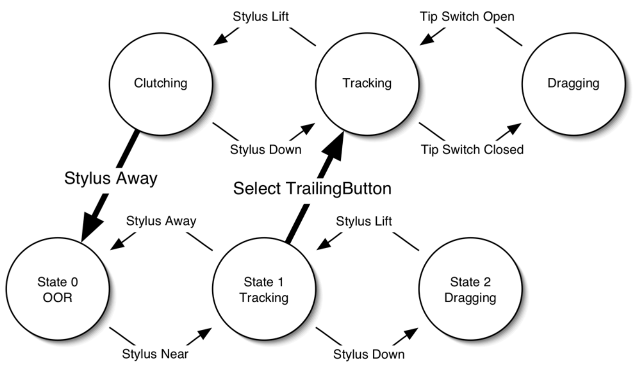

Figure 5: HybridPointing, allowing the user to work in absolute mode (*bottom*) or relative mode (*top*) with an easy means of switching between them.

Switching Between Absolute and Relative Input

We now give a more detailed explanation of the HybridPointing technique. The default mode is absolute input, so a user knowing nothing about our technique is able to use the pen in a familiar absolute manner. While in absolute mode, a trailing button "floats" nearby as it follows the pen (Figure 6a), its viscous dynamics keep it out of the way when pointing at targets, yet a quick, deliberate pen movement can trap it. To switch from absolute to relative mode, the user traps the trailing widget by quickly lifting the pen up and putting it down inside the circle (Figure 6b). Once trapped, the circle shrinks to a small size and stays immediately under the pen tip while the cursor is "disconnected" from the pen (Figure 6c) with a transition animation. The pen now controls the cursor in relative mode (Figure 6d).

To switch from relative input back into absolute input, the user lifts the pen away from the display past a predetermined threshold distance (Figure 6e). Once beyond the threshold, the circle is "released" from the pen tip and the cursor is brought under the pen (Figure 6f). The pen now controls the cursor in absolute mode again (Figure 6g).

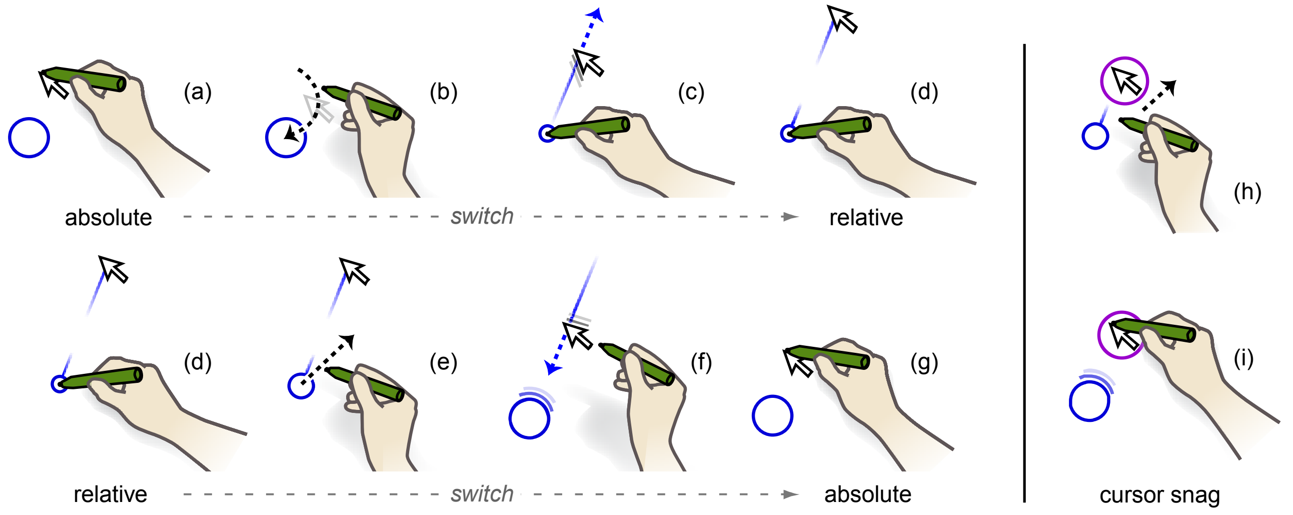

Figure 6: Switching between absolute and relative input. (*Top line*) When working in absolute mode, a quick click on the trailing button switches the system into relative input mode. (*Bottom line*) Lifting the pen far enough away from the display surface returns the system to absolute input mode. (*Right pair*) Alternatively, a user may switch from relative to absolute input by clicking on the graphical system pointer itself when in relative mode.

We had to choose the threshold distance carefully since the pen will be raised slightly off the display surface while clutching in relative mode. A small threshold may cause an erroneous switch back to absolute mode with relative clutching movements, but a large threshold will force the user to lift the pen an unnaturally far distance off of the display to indicate a switch. From pilot experimental data with pure relative input, we found that a threshold distance of 80mm provided the best tradeoff between erroneous switches and having to lift the pen too far.

While iterating our design we observed users occasionally attempting to click on the cursor when it was near the pen in relative mode. These users seemed to form a conceptual model which suggested that "snagging" the cursor was a reasonable way to switch from relative to absolute mode. To support this secondary switching method, we reveal a circular target around the cursor when the pen is near (Figure 6h), and clicking on the cursor returns to absolute mode (Figure 6i).

Visual Feedback

We implemented several means of visual and auditory feedback to communicate pen distance state, current input mode, and input mode switches.

To communicate pen distance state, the cursor fades out when the pen is off the surface (Figure 7b) and "dangles" [29] if the pen is far away (Figure 7c). This makes tracking (hover) and out of range states explicit to the user. In relative mode, the clutching action of the pen momentarily enters the hover state and then returns back to touching the display. This caused the cursor to flicker with each clutching action. A similar visual artifact occurred when switching from relative to absolute mode when the pen moved beyond the hover threshold. To eliminate these momentary visual distractions, we fade or dangle the cursor only after the state is unchanged for 500ms.

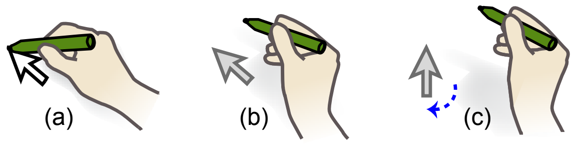

Figure 7. Pen distance visual feedback. (a) cursor is normal when pen is on display surface; (b) cursor fades out when pen hovers; (c) cursor "dangles" when pen moves out-of-range [29].

In our early designs, we expected the current input state (absolute or relative) to be obvious because of the spatial relationship of the pen tip and the cursor: if the cursor was directly under the pen, the mode was absolute; if the cursor was not under the pen, the mode was relative. However, we found that this was not the case since we observed users confused about which input mode they were in. To combat this, we added a partial line joining the pen position and the cursor when in relative mode (Figure 6d). This reminds the user which input mode they are in, and provides an added benefit in that the length and direction of the line help to locate the cursor – with relative input on a large display, users sometimes forget where the cursor is located.

In addition to making the current mode obvious, we also use audio and animation to indicate a mode switch. We draw attention to the movement of the cursor as it is "disconnected" from the pen during an absolute to relative switch by drawing an animated line resembling a comet trail from the pen to the cursor position., and similarly when the cursor is "connected" to the pen during a relative to absolute switch. We also play a distinctive, but subtle, musical chord each time a mode switch occurs with a major chord for a switch to relative and a minor chord for a switch to absolute.

EXPERIMENT

We conducted an experiment in which participants performed a target selection task using absolute input, relative input, or our hybrid technique. The goal of this study was to 1) compare users' pointing performance across the three techniques, and 2) observe whether or not participants would take advantage of input mapping switching in the hybrid condition, and in what situations they would do so.

Participants

We recruited 30 participants ranging in age from 17 to 37 years. These participants were students and were not paid for their participation, although participants were entered into a raffle for an Apple iPod.

Apparatus

We used a 5m wide, 1.8m high, back projected solid glass screen display (Figure 8), 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 [8].

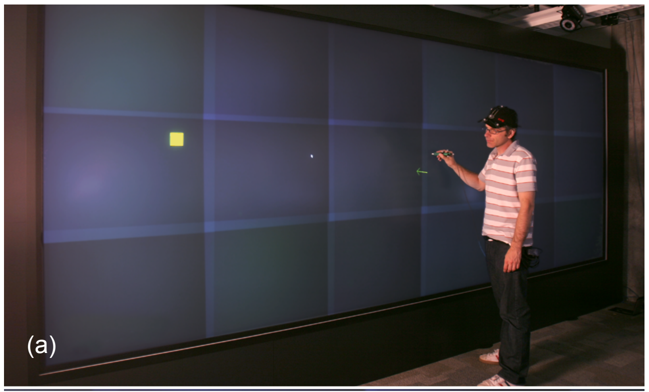

Figure 8: The wall sized display use in our study.

We used a Vicon [28] 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 [29].

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 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.

Task

A standard 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 yellow squares on a black background. When participants successfully clicked 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. Participants had to successfully click and release within a target before the next target would appear, even if this required multiple clicks. This effectively removes the possibility that participants may try to carelessly "race" through the experiment.

Design

We used a repeated-measures design with both between- and within-participant factors. The between-participant independent variable was *input mapping* (absolute, relative, and hybrid). The within-participant independent variables were *target width* (8, 16, 32, and 64 pixels), and *target distance* (946, 1892, 2838, and 3784 pixels).

Each participant 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 each *target width*. Experimental sessions lasted about 45 minutes.

Participants were divided randomly into three groups of ten. One group performed the experiment using the HybridPointing technique, the second group used an absolute *input mapping*, and the third group used the relative mapping. In summary, the design was:

10 participants (per input mapping technique) x

3 input mappings (absolute, relative, and hybrid) x

7 blocks per mapping x

4 target widths (8, 16, 32, 64 pixels) x

4 target distances (946, 1892, 2838, 3784 pixels) x

3 repetitions

= 10080 selections in total.

Results

It is traditional to use Fitts' law to model and analyze performance in pointing experiments. In our study, however, there are various changes in input styles and user body movements and positions that occur throughout the experiment, making it questionable as to whether Fitts' law is an appropriate model to use. Further, the value of creating Fitts' models is that they allow for cross-experiment and cross-device comparisons. Since ours was a single experiment using the same apparatus, with the simple goal of comparing three techniques against one another, there is little to be gained from a Fitts' law analysis.

Selection Time Analysis

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. 79 trials in which the selection time was greater than three standard deviations from a participant's mean selection time were counted as outliers and removed. These trials represent 1.8% of our data.

We expected to see strong learning effects, especially with the hybrid input technique. After removing the first 4 blocks, block no longer had a significant main effect on selection time ($F_{2,54} = 2.25$, $p = 0.12$). Thus, we considered only the last 3 blocks in the rest of our analysis.

Overall, *input mapping* did not have a significant main effect in terms of selection time, with mean selection times of 3.22s, 2.97s, and 3.31s for absolute, relative, and hybrid input respectively. As one would expect from Fitts' law, both *target width* and *target distance* had a significant effect on selection time ($F_{3,81} = 354.95$, $p < 0.001$ and $F_{3,81} = 359.63$, $p < 0.001$ for *width* and *distance* respectively), with smaller and farther targets taking longer to select.

Most interestingly, there was a significant interaction between *input mapping* and *target distance* ($F_{6,81} = 35.97$, $p < 0.001$). Figure 9 shows the mean selection times for each *target distance* for each of the three *input mappings*. Absolute input resulted in the fastest times for close targets, and relative input was the fastest for distant targets. The performance of hybrid input closely matched that of relative input, with a constant offset. No other significant interactions were observed for selection time.

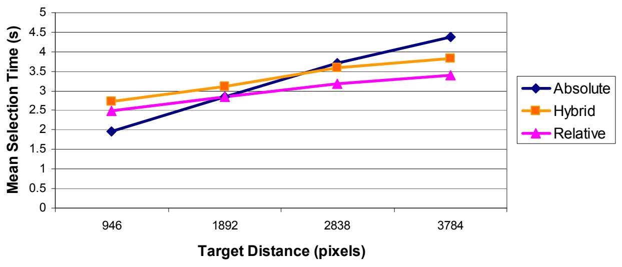

Figure 9: Mean selection times for each *target distance* for each of the three *input mappings*.

Selection Error Analysis

A repeated-measures ANOVA indicated that *input mapping* had a significant effect on selection error rate ($F_{1,27} = 4.01$, $p = 0.03$), with mean error rates of 4.2%, 3.9%, and 6.8% for absolute, relative, and hybrid input respectively. Figure 10 shows the mean error rates for each technique.

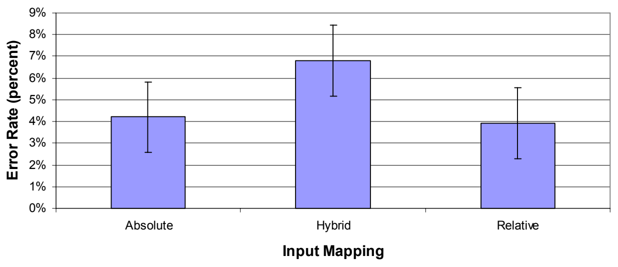

Figure 10: Mean error rates for each *input mapping*.

As one would expect, *target width* has a significant effect on error rate ($F_{3,81} = 67.93$, $p < 0.001$) with smaller targets being harder to select; however, there was little difference among error rates for differently distanced targets ($F_{3,81} = 0.36$, $p < 0.78$). No other significant effects or interactions were observed for selection error rate.

Hybrid Technique Analysis

A major goal of this experiment was to observe whether or not participants would take advantage of their ability to switch input mapping when using the hybrid technique, and in what situations they would do so. In order to explore this issue, we divided the hybrid trials from our study into five groups for further analysis.

Absolute-only Trials were trials in which the participant started in absolute input mode and remained in absolute mode until the target was selected. *Relative-only Trials* were those where a participant remained in relative input mode from target presentation to target selection. *Relative-to-Absolute Trials* indicated a switch from relative to absolute input, and *Absolute-to-Relative Trials* a switch in the other direction. Finally, *Multiple-Switch Trials* occurred when the participant made more than one input mapping switch during a single trial. Table 1 shows the number of each type of switches for the hybrid trials. These totals include both correct trials and trials in which the participant committed a selection error.

Trial Type	Count	Percent
Absolute-only Trials	48	3.3 %
Relative-only Trials	910	63.2 %
Absolute-to-Relative Trials	222	15.4 %
Relative-to-Absolute Trials	220	15.3 %
Multiple-Switch Trials	40	2.7 %
Total	1440	100 %

Table 1: Number and types of hybrid input trials.

The majority of trials were relative-only trials, and given the nature of our task, it was not a bad strategy to stand near the middle of the display and use relative input for all but the closest targets. The high number of relative-only trials probably explains the resemblance between the relative and hybrid lines shown in Figure 9. The 40 trials in which there were multiple-switches between absolute and relative input indicate one of two things occurred. Either a switch was made accidentally and then the user corrected their mistake, or the participant was unsure or confused as to which mapping was the best to use.

Figure 11 shows a histogram of the frequency of hybrid trial type for different distances between the physical pen and target at the start of a trial (as opposed to the distance between the graphical pointer and the target at the start of each trial). When this distance was small, participants were likely to switch from relative input to absolute input to select the nearby target. If the distance was small and they were already using an absolute mapping, they tended to remain in absolute mode. When the distance was large, participants either remained in relative mode, or switched from absolute to relative input – in fact, there were no trials in the final three blocks of our study in which a participant remained in absolute mode to select a target more than 2 meters away. *Relative-only* did not have many long distance trials because when using the relative technique, subjects were not standing directly in front of the target at the end of a trial but were likely to be positioned in the middle of the display, making the distance between the pen and target at the start of the next trial shorter. In general, participants tended to switch modes about 1/3 of the time, and appeared to do so when it would be most advantageous.

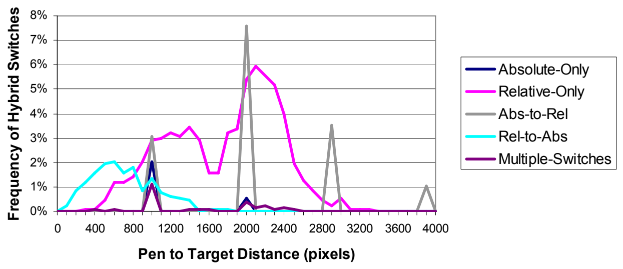

Figure 11: A histogram showing the frequency of hybrid trial type by the distance between the pen and the target at the beginning of a trial.

While the frequency of switching indicates that participants felt there was value to switching input mappings, one might ask how this switching affected their performance. Figure 12 shows the mean selection times for each *target distance* for each type of hybrid input trial. Again, there were no absolute-only trials for target distances over 2 meters. The performance of absolute-only hybrid input matched the performance of absolute input very closely. Similarly, the performance of relative-only hybrid input closely matched the performance of relative input trials – with a constant offset of about ¼ second. This penalty may be due to the need for the user to decide whether or not to switch input mappings in the hybrid technique. The greatest benefit in terms of selection time came from the relative-to-absolute transition for distantly spaced targets. The relatively flat line of the relative-to-absolute trials indicates that no matter where on the display the cursor was at the start of a trial, a nearby target could be selected in a short amount of time.

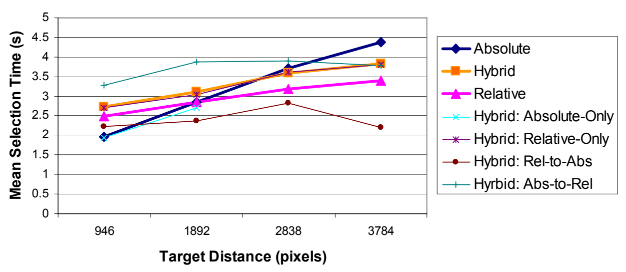

Figure 12: Mean selection times for each *target distance* for each of the three *input mappings* and each type of hybrid trial.

The higher error rate for hybrid trials as a whole compared to absolute and relative trials (Figure 10) prompted us to look at the error rates for each type of hybrid trial, which are shown in Figure 13. Many of the errors occurred in trials in which a participant made more than one switch between absolute and relative mapping. It seems as though participants who committed a switching error, as indicated by performing multiple-switches in a single trial, may have given-up on the trial and performed the target selection with a lower accuracy. The error rate of relative-only hybrid trials was very similar to the error rate of relative trials; however, the error rate for absolute-only hybrid trials was much higher than that of absolute trials. This difference may be due to the presence of the Trailing Widget in the absolute-only conditions, may be due to the management of multiple input mappings, or may simply be a difference between participants.

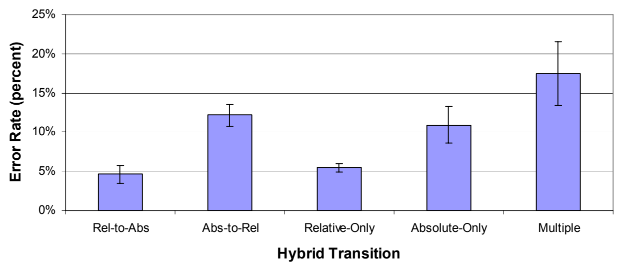

Figure 13: Error rates for each of the types of hybrid input trials.

Observations and Discussion

Our experimenter observed the seemingly contradictory fact that participants often had the most difficulty locating targets that were close to them. At the start of a trial, participants would first scan to the left or right, retrace his/her path if the target was not found, and only then take a step or two back from the display to check the areas immediately above or below their current position. When standing at arms length from the wall-sized display, the areas above and below the user's head are not easily visible, and one's arm and hand occlude portions of the display.

Several participants stated that the hybrid technique seemed to be a compromise between the absolute and relative mappings.

Interestingly, several participants felt that targets located high or low on the large display were most easily selected with relative input, even if they were very close to the participant.

While several participants initially accidentally activated the stylus tip-switch, either when selecting the trailing widget or when tracking in relative input mode, they improved with practice and eventually were able to drag the stylus across the screen without activating the switch.

Many participants commented on the visibility of targets on our large display, often complaining that distant, small targets were hard to see when standing close to the screen. One participant went so far as to suggest that we remove a column's worth of projectors to make the display smaller and thus more usable. Several felt that distant targets were difficult to select when in relative mode because they were harder to see, and several participants wanted to be able to back away from the display to get a better view. Finally, one participant felt that the system pointer should grow as it moves farther away from the user so that it is easier to see.

Practice seemed to greatly help participants manage relative input. By the end of the experiment, several participants had learned to reach all areas of the display without moving their feet. A few were able to reach even the most distant targets without clutching.

Several participants complained that it was too easy to accidentally switch from relative to absolute mode by moving their pen too far away from the display when clutching. While they learned to work within this threshold, a quick look at the study's logs indicates that individuals have different distances that they use to clutch, indicating that this distance threshold may be best implemented as an adjustable parameter.

Feelings about the trailing widget were mixed. Several participants felt that the widget was "in the way" or "distracting" at first, although all learned to work with it, eventually "ignoring it until it was needed". One participant suggested that the trailing widget be aware of target locations so that it could better position itself to stay out of the way. Many participants suggested other methods for switching between absolute and relative input, the most popular being including an extra button on the stylus to switch modes.

DESIGN VARIATIONS

Our experiment demonstrated the value of HybridPointing under certain conditions on a single large display. In this section, we describe other scenarios in which HybridPointing would be valuable, describe an extension to HybridPointing that allows for switching between absolute and relative input in the middle of a single dragging operation, describe other methods for switching between mappings, and describe how HybridPointing can be implemented on a variety of input devices with different sensing capabilities.

Multi–Display Environments

Multi-display environments, such as that described by Streitz et al. [27], have generated a lot of interest in recent years. These environments often include a heterogeneous mix of devices with different input capabilities. Johanson et al. presented a pointing technique, called PointRight [15], that allows a user to move a system pointer across the displays of multiple machines using a single relative input device. In the case of using a handheld Tablet in conjunction with a wall display or in sitting around a touch-sensitive table surrounded by large vertical displays. HybridPointing users could benefit from the strong affordance and performance of absolute input for close objects, while switching to relative input for distant ones.

HybridDragging

In addition to allowing the user to switch between periods of relative input to periods of absolute input, the addition of two more transitions allows users switch between mappings *in the middle of a single dragging operation*. The transitions between relative dragging and absolute dragging are particularly interesting. Moving from absolute to relative input while dragging an object allows a user to shoot an object off towards a distant part of the screen in a manner similar to the Go-Go interaction technique presented by Poupyrev et al. [22]. Conversely, when a distant target is selected in relative mode, it may be "vacuumed in" [5] and placed directly under the stylus. In both cases, the user can continue to work with the selected object without interruption.

HybridDragging with a Two-State Input Device

In the prototype system used in our experiment, a user controled the HybridPointer with a sophisticated multi-state input device. In this section, we detail how HybridPointing and HybridDragging can be used with input devices capable of sensing only two levels of input – a multi-touch sensitive tabletop. Figure 14 shows the two-state model for input with this device. Through giving the details of HybridPointing/Dragging with a second input device, we hope that the interested reader can understand how HybridPointing might be adapted to the specifics of their favorite input device – remembering that HybridPointing only needs an absolute mapping, a relative mapping, and a means of switching between them.

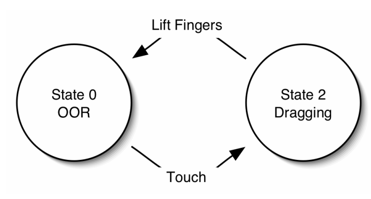

Figure 14: The two-state model for our touch-sensitive table.

Most two-state touch-sensitive input devices do not track the user's finger or stylus when they are above the input surface, and only sense contact with the device. This lack of tracking on the part of the system is easily overcome by the user, they simply track the physical stylus tip or their own finger, and the location at which input will occur when contact is made is obvious.

A laptop trackpad may be the most common two-state input device, and this device works in a relative manner. Normally, touches result in relative movement of the system pointer; however, a tap-and-then-drag on a trackpad starts a relative dragging operation. We model relative input with our two-state touch-sensitive table after relative input with a laptop trackpad.

Figure 15 shows a state transition model for HybridDragging on a two-state touch sensitive table. The default mapping is absolute, and a user sitting at this table for the first time will find the table behaving in the expected fashion, with input occurring directly under their finger. More advanced users may manage the switching between absolute and relative input with a touch from a second finger, in a manner similar to that described by Esenther and Ryall [9]. Touching with two fingers and then immediately lifting one will switch the system into relative tracking mode. If the user is already dragging an item with their index-finger in absolute mode, a tap with their thumb will switch them into relative mode. Unlike with a laptop trackpad, this model supports clutching while dragging – the dragged object remains selected until a second tap drops it at its current location.

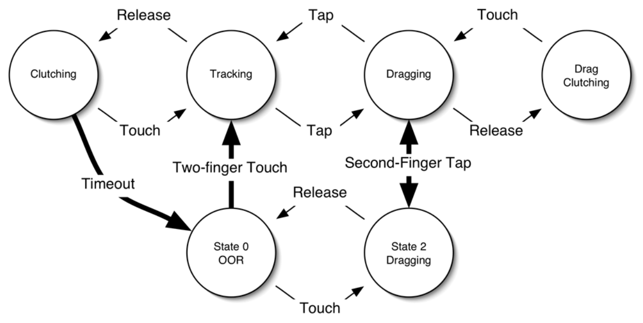

Figure 15: HybridDragging with a two-state touch sensitive tabletop. Users can switch between absolute and relative input in the middle of a single dragging operation.

CONCLUSION

HybridPointing is a viable alternative to working solely with either absolute or relative pen input. We demonstrated this technique with an implementation for a large, wall-sized display that includes the use of visual feedback to aid the user's understanding of input state. This feedback includes a new type of graphical element, the *trailing widget*, which keeps GUI elements "close at hand" on large displays where fixed elements may be out of reach. While our trailing button provided a convenient means of switching input mapping, a full exploration of trailing widgets is left for future work. Finally, we hope that the presentation of HyrbidPointing, and its extension HybridDragging, will allow the interested reader to implement their own version of our technique on their own large display or in a multi-display environment.

ACKNOWLEDGMENTS

We gratefully acknowledge our study participants and members of MERL and the DGP lab for their valuable discussion. We also thank Clarissa Mak, who ran many of our experimental sessions.

REFERENCES

1. Accot, J. and Zhai, S. Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. in *Proc. of ACM SIGCHI Conference on Human Factors in Computing Systems,* (Pittsburgh, PA, 1999), 466-472.
2. Baudisch, P., Cutrell, E., Hinckley, K. and Gruen, R., Mouse ether: accelerating the acquisition of targets across multi-monitor displays. in *CHI '04 extended abstracts*, (Vienna, Austria, 2004), 1379-1382.
3. Baudisch, P., Cutrell, E. and Robertson, G., High-Density Cursor: A Visualization Technique that Helps Users Keep Track of Fast-Moving Mouse Cursors. in *Proceedings of the ninth IFIP TC13 international conference on Human-Computer Interaction – INTERACT2003*, (Zürich, Switzerland, 2003), Springer, 236–243.
4. 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. in *Proceedings of the ninth IFIP TC13 international conference on Human-Computer Interaction*, (Zürich, Switzerland, 2003), 57–64.
5. Bezerianos, A. and Balakrishnan, R., The vacuum: facilitating the manipu-lation of distant objects. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, (Portland, Oregon, USA, 2005), ACM Press, 361-370.
6. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. 1993. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and interactive Techniques* SIGGRAPH '93. ACM Press, New York, NY, 73-80.
7. Buxton, W., A three-state model of graphical input. in *Proc. of the IFIP TC13 Third Interantional Conference on Human-Computer Interaction*, (North-Holland, 1990), 449-456.
8. Chromium. http://chromium.sourceforge.net.
9. Esenther, A. and Ryall, K. Fluid DTMouse: Better Mouse Support for Touch Based Interactions. in *Proc. of Advanced Visual Interfaces* (Venezia, Italy, May 23 - 26, 2006), ACM Press, New York, NY, 112-115.
10. Fitzmaurice, G., Khan, A., Pieke, R., Buxton, B., and Kurtenbach, G. (2003). Tracking menus. *ACM Symp. on User Interface Software and Technology*. p. 71-79.
11. Forlines, C., Vogel, D., Kong, N., Balakrishnan, R., Absolute vs. Relative Direct Pen Input. Mitsubishi Electric Research Labs Tech Report, TR2006-066.
12. Goldberg, D. and Goodisman, A. Stylus user inter-faces for manipulating text. in *Proc. of the ACM Symposium on User Interface Software and Technology*, (Hilton Head, SC, USA, 1991), 127-135.
13. Graham, E. and MacKenzie, C. Physical versus virtual pointing. in *Proc. of the CHI '96 Conf. on Human Factors in Computing Systems*, (Vancouver, BC, Canada, 1996), 292-299.
14. Grossman, T., Hinckley, K. Baudisch, P., Agrawala, M., Balakrishnan, R. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices. To appear in *Proceedings of CHI 2006*, Montreal, Canada, April 2006.
15. Johanson, B., Hutchins, G., Winograd, T., Stone, M., PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. in *Proc. of the ACM Conf. on User Interface and Software Technology*, (Paris, France, 2002), 227-234.
16. MacKenzie, I.S. and Oniszczak, A., A comparison of three selection techniques for touchpads. in *Proc. of the SIGCHI conference on Human factors in computing systems*, (Los Angeles, California, United States, 1998), ACM Press, 336-343.
17. Microsoft. Pointer ballistics for Windows XP. Accessed on Mar 23, 2006, http:*//www.microsoft.com/whdc/device/input/pointer-bal.mspx.*
18. Meyer, S., Cohen, O. and Nilsen, E. Device comparisons for goal-directed drawing tasks. in *Extended Abstracts of the 1994 Conference on Human Factors in Computing Systems - CHI '94*, (Portland, Oregon, USA), ACM Press, 251-252.
19. Myers, B., Bhatnagar, R., Nichols, J., Peck, C.H., Kong, D., Miller, R. and Long, C., Interacting at a distance: measuring the performance of laser pointers and other devices. in *Proc. of ACM CHI Conference on Human Factors in Computing Systems*, (Minneapolis, Minnesota, USA, 2002), 33-40.
20. Nacenta, M.A., Aliakseyeu, D., Subramanian, S. and Gutwin, C., A comparison of techniques for multi-display reaching. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, (Portland, Oregon, USA, 2005), 371-380.
21. Parker, J.K., Mandryk, R.L. and Inkpen, K.M., TractorBeam: seamless integration of local and remote pointing for tabletop displays. in *Proceedings of the 2005 conference on Graphics interface*, (Victoria, British Columbia, 2005), Canadian Human-Computer Communica-tions Society, 33-40.
22. Poupyrev, I., Billinghurst, M., Weghorst, S. and Ichikawa, T., The go-go interaction technique: non-linear mapping for direct manipulation in VR. in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, (Seattle, Washington, United States, 1996), 79-80.
23. Rekimoto, J. and Saitoh, M.,. Augmented Surfaces: A spatially continuous work space for hybrid computing environments. in *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI'99*, (Pittsburgh, Pennsylvania, USA, 1999), 378-385.
24. Sears, A. and Shneiderman, B. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. *International Journal of Man-Machine Studies*, 34(4). 593-613.
25. Shen, C., Everitt, K., Ryall, K., UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. in *Proceedings of the Fifth International Conference on Ubiquitous Computing*, (Seattle, Washington, USA, 2003), 281-288.
26. Shneiderman, B. Touchscreens now offer compelling uses. *IEEE Software*, *8* (2). 93-94,107.
27. Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R.., i-LAND: An interactive Landscape for Creativity and Innovation. in *Proc. of the ACM Conf. on Human Factors in Computing Systems*, (1999), 120-127.
28. Vicon. http://www.vicon.com
29. Vogel, D. and Balakrishnan, R., Distant freehand pointing and clicking on very large, high resolution displays. in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, (Seattle, WA, USA, 2005), 33-42.