

# Pointing Lenses: Facilitating Stylus Input through Visual- and Motor-Space Magnification

Gonzalo Ramos<sup>1</sup>, Andy Cockburn<sup>2</sup>, Ravin Balakrishnan<sup>1</sup>, Michel Beaudouin-Lafon<sup>3</sup>

<sup>1</sup>University of Toronto  
bonzo, ravin@dgp.toronto.edu

<sup>2</sup>University of Canterbury  
andy@cosc.canterbury.ac.nz

<sup>3</sup>Univ. Paris-Sud & CNRS, INRIA  
mbl@lri.fr

## ABSTRACT

Using a stylus on a tablet computer to acquire small targets can be challenging. In this paper we present pointing lenses – interaction techniques that help users acquire and select targets by presenting them with an enlarged visual and interaction area. We present and study three pointing lenses for pen-based systems and find that our proposed Pressure-Activated Lens is the top overall performer in terms of speed, accuracy and user preference. In addition, our experimental results not only show that participants find all pointing lenses beneficial for targets smaller than 5 pixels, but they also suggest that this benefit may extend to larger targets as well.

## Author Keywords

Design, Experimentation, Human Factors.

## ACM Classification Keywords

H.5.2 [User Interfaces]: Interaction styles; I.3.6 [Methodology and Techniques]: Interaction techniques

## INTRODUCTION

Acquiring and selecting targets are common and fundamental tasks in current GUIs. However, small targets or the characteristics of the input device being used can make these tasks difficult. There are many scenarios in modern GUIs where users need to acquire small targets. For example, draggable window borders, handles, and divider bars can be smaller than 5 pixels. Placing of the “I-beam” cursor during text editing or selection tasks can demand great accuracy – i.e., at 100% zoom, many Times 10pt characters, are less than 5 pixels wide. Footnotes, subscripts, and inter-character spacing are even smaller. Direct-pointing scenarios involving touch screens and pen-computers further introduce a number of factors that influence the acquisition and selection of targets. Traditional GUIs when translated to portable, small form factors such as phones, PDAs or TabletPCs often offer users particularly small targets. Also, the nature of the input device in these cases cannot be ig-

nored. A finger can be a versatile and expressive instrument, but a coarse pointing device. For example, the effective size of radio buttons and similar small widgets is further reduced if one uses a finger on a touch-screen. In the context of tablet devices, the effective size of small targets can be further reduced by the impaired accuracy due to visual parallax and the display's occlusion by the stylus, factors that make targeting tasks challenging by misleading users as to the true location of the cursor. Because of their steadily growing presence, we focus on pen-based interaction and scenarios, where it is crucial to have the means to assist users performing difficult selections. Furthermore, we seek a system-based solution that would not require existing platforms to be re-engineered or applications to be rewritten. As such, the results of our research could be applied to both existing and new platforms and applications.

In this paper, we present *pointing lenses*, an interaction technique that temporarily enlarges both the visual and motor area under a pointer's cursor (Figure 1). This enlarged area provides users with visually magnified targets that also virtually reduce the control-display (CD) ratio of the input device, facilitating pointing tasks. We present three types of pointing lenses that help users acquire and select targets. We then investigate the performance of these lenses relative to the condition of selecting targets without any lens. We also present and discuss the results of a study that observes the frequency of use of pointing lenses for targets of different sizes. Finally, we elaborate on additional design aspects of pointing lenses and present topics for future work.

ing a stylus on a tablet computer to acquire small targets can be challenging. In this paper we present pointing lenses – interaction techniques that help users acquire and select targets by presenting them with an enlarged visual and interaction area. We present and study three pointing lenses for pen-based systems and find that our proposed Pressure-Activated Lens is the top overall performer in terms of speed, accuracy and user preference. In addition, our experimental results not only show that participants find all pointing lenses beneficial for targets smaller than 5 pixels, but they also suggest that this benefit may extend to larger targets as well.

## Author Keywords

Submitted to CHI2007

Figure 1: A pointing lens used to magnify the inter-character space in a standard word processor session.

## RELATED WORK

Fitts [7] offers the seminal work on the study of pointing tasks as a basic operation. In turn, knowledge from this work has inspired many methods and techniques that seek to facilitate the acquisition and selection of targets. Most of these techniques reduce the Fitts' *Index of Difficulty*, generally by increasing a target's size. McGuffin et al. [12] show that increasing a target's size can be beneficial even at the final stages of a pointing task. Semantic pointing [5] improves performance by dynamically adjusting CD ratio as the cursor approaches the target, with the level of adaptation depending on the target's importance. While effective, semantic pointing requires a priori knowledge of the targets and their importance. Grossman et al. [11, 18] propose the bubble cursor, a target acquisition technique based on area cursors [11, 18]. The bubble cursor dynamically changes its size depending on the proximity of surrounding targets so as to encompass only one target at any time. However, bubble cursor's performance degrades in densely packed target-rich environments often seen in current user interfaces. Like semantic pointing, the bubble cursor also needs knowledge about the size and location of the targets in the interface.

Precise pointing tasks can be difficult with an imprecise input device such as a finger used on a touch-screen. Potter et al. [14] propose take-off, a technique to increase the accuracy of touch screen pointing. With the take-off technique, a target's location is defined by the point at which the finger is lifted. This technique allows users to adjust a cursor's position by keeping their finger in contact with the screen. However, take-off does not always facilitate the acquisition of small targets, since it does not allow for the adjustment of the CD ratio, or the target area's zoom factor. Albinsson and Zhai [1] also explore different interaction designs for the selection of small targets in touch-screen environments. All their methods require users to perform a series of steps to do a selection. In particular the most successful technique in their studies, ZoomPointing, required users to first define an area of interest to be magnified and then to select a target within it. Benko et al. [3] seek to develop more fluid precise-selection techniques, leveraging the capabilities of multi-touch screens. Their best performer, the Dual Finger Stretch, uses a person's non-dominant fingertip to define a zooming area centered at the primary finger, which then performs the selection. Forlines et al. [9] address similar issues of imprecise, noisy input control in the context of hand-held projectors. Their Zoom-and-Pick technique lets users fluidly zoom-in on areas of interest and permits precise target acquisition and selections. Zoom-and-Pick leverages the many degrees of freedom available from a hand moving in 3D space. In particular, it uses the roll action of the hand-held projector to both invoke a fisheye lens and control its zoom level. While this technique has the potential to be applied system-wide, it requires a unique input device with additional degrees-of-freedom beyond spatial x-y cursor positioning. Ramos and Balakrishnan's [15] Zliding technique leverages the pressure-sensing capabilities of current TabletPCs to enable

users to fluidly acquire very small targets. While the authors discuss the possibility of using Zliding coupled with a magnifying lens to acquire very small targets in the GUI, they do not provide further details or usability information.

These previous efforts help guide our goal of providing a system-wide method that fluidly magnifies the user's area of interest and ultimately facilitates pointing tasks for both regular and small targets.

## POINTING LENSES

The concept of magnifying lenses in the GUI is not new. Some window managers provide accessibility features that offer users a virtual magnifying lens that often consists of a rectangular region where a set of corresponding screen pixels are shown at a larger scale. Toolglasses and MagicLenses [4] also provide a magnifying lens that assists with pointing tasks. Still, accessibility lenses are highly modal and provide no motor-space enlargement. Thus, while they are helpful for users with visual impairments, they do not assist in the motor actions of target selection. MagicLenses typically require a bimanual interaction for invocation and use, making it practically infeasible for portable pen-based computers. In contrast to these past designs, the main goal of our pointing lenses is to facilitate quick and fluid target acquisition, with both visual- and motor-space enlargement, using a single pointing device. Consequently, we propose a set of design guidelines for pointing lenses:

- *Location*: The lens should be located in close proximity to the area being magnified.
- *Activation*: Activating the lens should be a fluid, lightweight operation (e.g., not require trips to toolbars).
- *Control*: Users should be allowed to adjust the lens' position, as well as to manipulate the lens' zoom factor.
- *Zero-knowledge*: The lens should not require knowledge of the interface's semantics underneath it and should treat the magnified area as a bitmap (or vector image).
- *Legacy*: When the pointing lens is not active or being used, users should be able to engage the interface normally – i.e., a click is a click, a drag is a drag.

In the following sections we present three design variations of pointing lenses, all of which offer users access to an enlarged visual and motor area that corresponds to a particular region of interest on the display. For now, we consider the lenses' magnification factor as a fixed parameter but will explore methods for adjusting it in later sections. In our current implementations we use a zoom factor of 4 and a lens measuring 128 by 128 pixels.

### Delay-Activated Lens

The Delay-Activated Lens (DAL) appears when users dwell over a region of interest with the stylus for a predetermined number of milliseconds. This activation method is common in many GUIs and reveals additional information about a particular element in the interface – e.g., tool tips. However,

in this case the timeout has to be carefully chosen to permit normal pointing interaction and to keep false activations to a minimum. A very short timeout value will cause too many false activations, while a long timeout value will make the interaction slow. We found through early pilot studies that a value of 400ms resulted in acceptable performance.

The delay-activated lens screen-grabs a prescribed square area underneath the pointer and presents it as a magnified square-shaped overlay. The lens appears centered at the stylus' location and is displayed as a semi-transparent surface to avoid occlusion of the area underneath (Figure 2). Once active, users can interact with the pixels on the lens the way they would normally interact with any graphical user interface. We use a "road markings" (Figure 2) metaphor to communicate graphically the lens' behavior when the pointer gets close to one of the lens' edges. We use an inner solid line/outer dashed line to indicate that it would be a "violation" to cross this boundary from inside the lens.

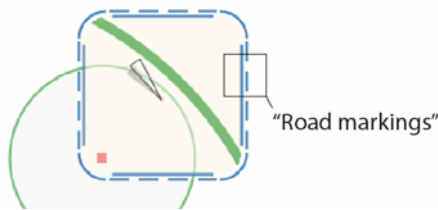


Figure 2: Detail of a delay-activated lens with "road markings" on its edges.

When users reach the edges of the lens at low speeds, the lens follows the stylus in a tracking behavior similar to tracking menus [8] (Figure 3). This allows for adjustments of the lens focus as if it were a sliding window over a magnified visual and motor space. As the stylus moves, a *cursor beam* connecting the stylus' position to the cursor's corresponding position in normal screen space is displayed. Finally, if a user approaches a lens' edge at a fast speed, the boundary is "violated" and the lens is dismissed (Figure 4). The lens can also be dismissed by lifting the stylus outside the interactive display's tracking region.

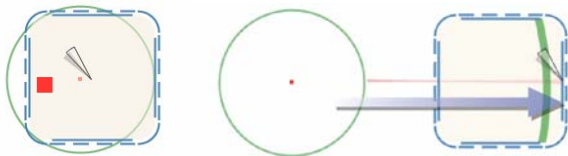


Figure 3: (left to right) The stylus pushes the lens if it reaches the lens' edges slowly. (right) A cursor beam connects the stylus with the cursor's true location.



Figure 4: Lens dismissal. A stylus that moves quickly from inside to out "breaks" the lens' boundaries and dismisses it.

### Pressure-Activated Lens

The Pressure-Activated Lens (PAL) becomes active when the pressure applied by a user through the stylus is greater than a predefined threshold (in our implementation we use 60% of the maximum value sensed by our Wacom tablet). As the applied pressure increases, the lens fades in over the area of interest. This gradual appearance provides a visual cue as to the effects of increasing pressure. In addition, a pressure cursor [15] offers further feedback as to the amount of pressure applied through the stylus (Figure 5). Apart from the activation method, the pressure-activated lens looks and behaves exactly like the delay-activated lens. We believe this difference reduces the likelihood of false activations. Also, the use of pressure offers users more control over the activation speed. Once the lens is activated, the stylus can be lifted from the tablet surface, whereupon the lens behaves just like the delay-activated lens.

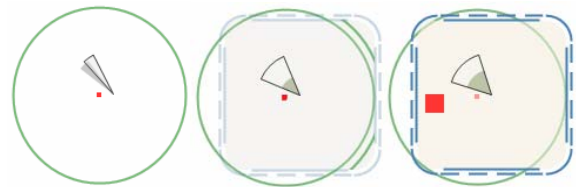


Figure 5: Pressure-Activated Lens. (left to right) As the pressure applied through the stylus increases, the lens gradually appears. After a threshold is reached, the lens becomes active.

### Trailing Lens

The Trailing Lens (TL) is always active when the stylus is within tracking distance of the tablet surface and it follows the stylus' position at a prescribed location and distance (Figure 6). By default, the trailing lens is displayed at the lower left of the cursor, but we allow users to define its position through a configuration dialog box, for example to accommodate left-handed users. This lens is visually similar to the pressure-activated and delay-activated lenses and is displayed as a square area connected to the cursor's current position through a *lens beam* (Figure 6).

The lens follows the stylus in a movement that exhibits a viscous behavior similar to the trailing widget described by Forlines et al. [10] – i.e., the lens follows the stylus' movements with a certain delay. While this viscosity pulls the lens towards, or pushes it away from, the stylus' current position, it also allows users to "trap" or "catch" the lens by quickly moving the cursor into the lens area (Figure 6).

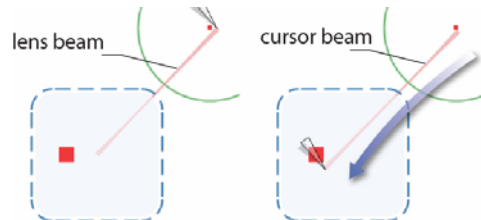


Figure 6: (left) The trailing lens magnifies the area beneath a cursor and presents it at a slight distance away. (right) Users can "trap" the lens by quickly moving the cursor into its area.

Once trapped, users can interact with the pixels inside the trailing lens' area the same way they would normally interact with other interface elements. Similar to the delay-activated and pressure-activated lenses, we show a *cursor beam* that connects the stylus' position to the cursor's position on normal screen space. We use a "road markings" metaphor to express graphically the trailing lens' behavior when the stylus gets close to the lens' edges. We use a dashed line to indicate that one is allowed to cross this boundary while inside or outside the lens. This boundary is crossed inwards when the user catches the lens. Conversely, the lens reverts to its trailing behavior when the user moves the stylus outside the lens' area. Finally, the lens fades away from the screen if one lifts the stylus outside the interactive display's tracking region.

### Dragging and Cross-Scale Dragging

Like pointing, dragging is a predominant interaction present in GUIs. Our three pointing lenses naturally incorporate dragging operations within a particular magnification factor or scale – i.e., dragging occurs fully inside the lens or fully outside the lens. There are occasions where this is the desired behavior – e.g., rubber-banding a small area on a drawing. Nonetheless, there are other occasions where a user wants to perform a dragging action that occurs across scales – e.g., targeting the edge of a window (targeting and pointing in an enlarged space) and resizing it (dragging in normal space). We call this type of interaction cross-scale dragging. Our current design of pointing lenses permits cross-scale dragging tasks. We use the speed at which one approaches the lens' edge while dragging to switch from enlarged space to normal space. In other words, if we reach a lens' edge slowly, the lens is pushed. If instead we approach the lens' edge quickly, the lens disappears and we switch to normal space (Figure 7). While in normal space, we can switch back to enlarged space using a lens' own activation method – i.e., dwelling, pressure, or trapping gesture (Figure 8).

Although we discuss dragging, the focus of the current paper is on the more elemental and ubiquitous pointing task. As such, a full evaluation of either same- or cross-scale dragging is beyond the scope of this paper.



**Figure 7: Cross-scale switching.** (left to right) One crosses the lens' edge at fast speed to switch from enlarged space to normal space.



**Figure 8: (left to right) One uses the lens' activation method to go back to enlarged space (here we see the pressure-activated lens)**

## EXPERIMENT 1: LENS' ACTIVATION AND ACCESS

The primary difference between our three lens designs is their activation mechanism. To evaluate possible differences in terms of activation and access time of these lenses, we conducted a quantitative study. The study contrasts the three lenses and the case where no lens is available in an elemental pointing task. In particular, we are interested in exploring the tradeoffs between the different lenses' activation techniques. It is not obvious which lens will be faster: holding the stylus still for a prescribed (and sensible) period of time, a variation in the pressure applied through the stylus, or a quick flick of the stylus. Also, any advantage seen in a particular technique might be negated by users having to visually search for an enlarged target in the lens' area. Finally, we want to determine the threshold target size at which using any of the pointing lens techniques is slower than using no lens at all.

### Apparatus

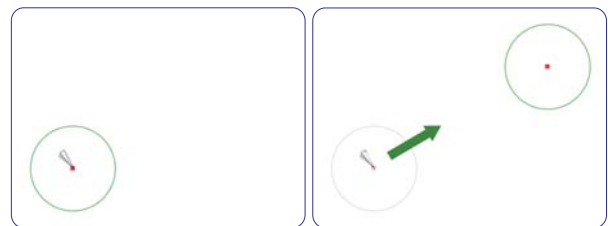
We used a Wacom Cintiq SX18 18" diagonal interactive display at a 1280 x 1024 pixel resolution. Participants interacted on the display's surface through a wireless stylus with a pressure-sensitive tip. The interactive display was set flat on a desk, as if it were a TabletPC in slate configuration.

### Participants

Twelve volunteers, 18-44 years old participated in the study and were recruited from the University of Toronto population through e-mail lists. We provided no compensation.

### Task and Stimuli

A sequential target acquisition task was used. At the beginning of a trial, a single red square target would be displayed (Figure 9-left). After that target was selected by tapping on it with the stylus, it was replaced by an arrow pointing to the next target (Figure 9-right), which appeared at a random location on the screen, but at one of two prescribed distances (short or long) from the previous target. We displayed targets with a surrounding green halo (Figure 9) to improve their visibility and reduce visual search times. This process was repeated for a sequence of 10 target selections within each experimental trial.



**Figure 9: Trial stimuli.** (left) Screen before a target is selected. (right) After a target is selected, it is replaced by an arrow pointing in the direction of the next target.

### Procedure and Design

We used a 4 technique (*none, DAL, PAL, TL*) × 4 target size (*1, 2, 4, 8 pixels*) within-participants design. The dependent variables were *selection time* and *errors*. We computed

selection time as the time elapsed between the moment a target appears on the screen for the first time and the moment the user successfully selects it. We recorded as errors the number of times users missed a target. Since one could only advance to the next target after successfully selecting the previous one, users were motivated to perform well. For this experiment, selections needed to occur using the given technique for that trial— i.e., for the three lens conditions, the lens needed to be activated and the target selected within its enlarged area. We enforce this condition to effectively measure the cost of activating and working within the expanded region of a lens. In order to ensure that participants did not get too frustrated with multiple erroneous selections of particularly difficult to select targets (i.e., the smallest ones), we considered a trial to be a complete failure when the user missed the target more than ten times. In this case, the current target was dismissed and the next appeared, resuming the normal flow of the experimental trial. In our experiment, such complete failures only occurred for the no-lens condition when the target’s size was 1-pixel (25% failures) and 2-pixels (1.7% failures).

For each technique we asked participants to complete 4 blocks of trials. Each block consisted of 4 trials (one per target size) repeated twice. In turn, each experimental trial consisted of 10 target selection tasks as described previously. The two possible distances between two consecutive targets were evenly distributed across trials and blocks. As a result, on average, the same number of short and long distances was traversed when going from one target to the next. The presentation of trials within a block was randomized. The presentation order of the techniques was balanced using a Latin square and was considered as a between-subjects factor. In summary, the experiment consisted of:

- 12 participants × 4 techniques × 4 blocks × 4 sizes ×
- 10 targets × 2 repetitions - 4 techniques × 10 warm-up selections = 15320 selection tasks.

Prior to the first use of each technique, we explained to the participants the way it worked. We instructed participants to be as quick and accurate as possible and only to take breaks between experimental trials. At the end of the experiment we asked participants to rank the techniques and to provide qualitative feedback.

### Results

The study took an average of 1 hour per participant. We performed a 4 (technique) × 4 (target size) × 4 (block) repeated measures analysis of variance (RM-ANOVA) on the logarithmically transformed selection times and the errors. Logarithmically transforming selection times corrects for the skewing often present in human response data, and reduces the influence of outliers. The presentation order of the techniques had no effect on the selection times or the errors. Because we are interested in how each technique performs for different target sizes, we examine technique × size interactions using Bonferroni-corrected pairwise comparisons.

### Selection Time

We found a main effect for technique ( $F_{3,15}=34.33$ ,  $p<0.0001$ ), where the PAL resulted in fastest selections. Pairwise comparisons reveal that PAL was significantly faster than using no lens ( $p=0.001$ ), DAL ( $p=0.013$ ) and the TL ( $p<0.0001$ ). In turn, pairwise comparisons reveal no significant differences between no lens, the DAL and the TL techniques ( $p>0.132$ ) (Figure 10).

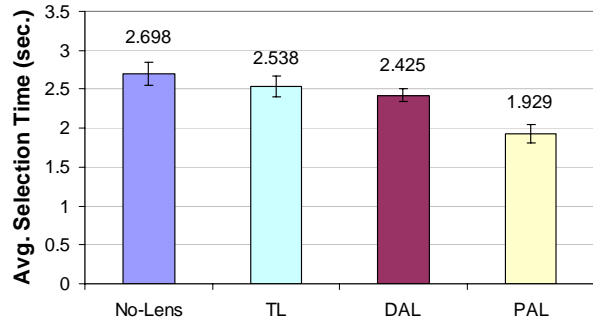


Figure 10: Average selection time (+/- SE) per technique.

As expected, there was a main effect for target size ( $F_{3,15}=619.08$ ,  $p<0.0001$ ). Pairwise comparisons show that selection times decrease ( $p<0.009$  for all sizes) as the target’s size increases. We also observe a significant technique × size interaction ( $F_{9,45}=97.62$ ,  $p<0.0001$ ). Post-hoc pairwise comparisons reveal different patterns for each of the target’s sizes. For 1-pixel targets users were significantly faster with the PAL than with no-lens ( $p<0.0001$ ), the DAL ( $p=0.028$ ) and the TL ( $p<0.0001$ ). For 2-pixel targets the PAL was still the fastest technique, but only significantly so to the no-lens ( $p=0.003$ ) and the TL ( $p=0.001$ ) conditions. For 4-pixel targets the average selection time was still lowest with the PAL technique; however it was only significantly faster than the DAL ( $p=0.009$ ) and the TL ( $p=0.01$ ). For this target’s size, no-lens was only significantly faster than the TL ( $p=0.033$ ) technique. For 8-pixel targets, selections were the fastest using no-lens, but it was only significantly faster than the DAL ( $p=0.001$ ), and the TL ( $p=0.007$ ). The PAL was the second fastest technique for this target’s size and also significantly faster than the DAL ( $p=0.012$ ) and the TL ( $p=0.028$ ) (Figure 11).

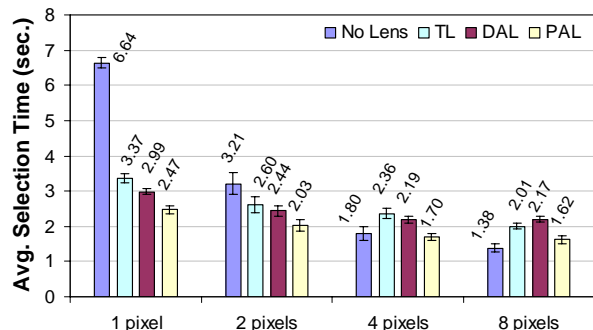


Figure 11: Average selection time (+/- SE) per size, technique.

There was a main effect for block ( $F_{3,15}=71.8$ ,  $p<0.0001$ ). However, post-hoc pairwise comparisons reveal that for each technique individually, only the first block of trials was significantly different from the rest. We believe this is due to participants quickly becoming familiar with the techniques. For each technique, we did not observe significant differences across blocks 2, 3, and 4. Further post-hoc pairwise comparisons reveal that this trend of no significant learning effects after the first experimental block is repeated for each target size condition. While there was an effect for the *short* and *long* distances, we observed no distance  $\times$  technique interaction.

Our results indicate that in terms of selection time, pointing lenses stop being beneficial for targets larger than 4 pixels. In addition, for the cases where it is beneficial to have a pointing lens, the PAL technique is the quickest to use.

### Errors

We use lower bounds for the errors results, because the data does not pass Mauchly's sphericity test. We found a main effect for technique ( $F_{1,7}=41.33$ ,  $p<0.0001$ ). Users made the fewest errors with the DAL, which was only significantly more accurate than the no-lens condition. As expected, we observed a main effect for target's size ( $F_{1,7}=56.385$ ,  $p<0.0001$ ) (Figure 12).

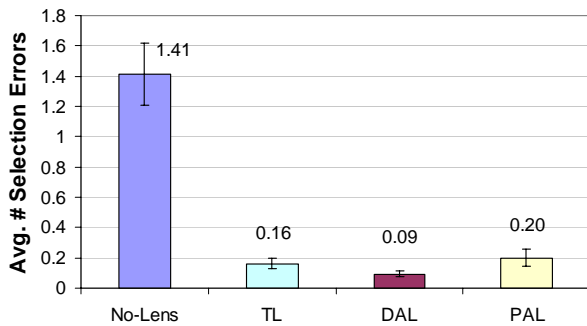


Figure 12: Average # of errors (+/- SE) per technique.

Pairwise comparisons show how participants were significantly more accurate as the target's size increased from 1, 2, to 4 pixels ( $p<0.02$ ). While there were fewer errors selecting targets of size 8 than selecting targets of size 4, this difference was not significant ( $p=0.119$ ) (Figure 13).

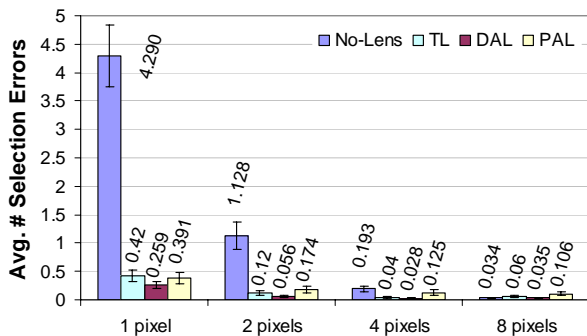


Figure 13: Average # of errors (+/- SE) per size and technique.

Post-hoc comparisons allow us to see the differences in accuracy between techniques at different target sizes. For 1- and 2-pixel targets users made significantly more errors when making selections without a lens than when using either the DAL ( $p<0.0001$ ), PAL ( $p<0.002$ ) or the TL ( $p<0.0001$ ). For these target sizes, the errors made with the three pointing lens techniques were not significantly different. For 4- and 8-pixel targets there were no significant differences between any of the four techniques (Figure 13).

Our post-hoc analysis reveals a significant cross-block improvement in accuracy for the no-lens condition, where we observed significant differences between the last and both the first ( $p=0.08$ ) and second ( $p=0.037$ ) experimental blocks. There were no effects for block for the three pointing lens techniques. Thus, unlike what we observed on the selection time data, pointing lenses are beneficial in terms of accuracy for all levels of the target's size.

### Qualitative Feedback

We asked participants to rank the ease of target selection with each of the techniques using a 7-point Likert scale ranging from 0 (strongly disagree) to 6 (strongly agree). The PAL ranked highest (mean=5.36, SE=0.14) (agree / strongly agree), followed by the DAL (mean=4.81, SE=0.16) (somewhat agree / agree), TL (mean=4.09, SE=0.37) (somewhat agree) and no-lens (mean=0.95, SE=0.27) (disagree). We asked participants to rank the techniques in order of preference from 1 (best) to 4 (worst). Eleven participants answered producing the following results: the PAL ranked top (mean=1, SE=0), followed by the DAL (mean=2.63, SE=0.2), the TL (mean=2.54, SE=0.15) and no lens (mean=3.82, SE=0.18). Participants' preferences are consistent with our quantitative results – i.e., the PAL was the preferred technique.

Participants commented that the PAL gave them a comfortable sense of control over the interaction in direct contrast to the DAL technique. Furthermore, some participants commented that they quickly became familiar with the “double tap” required to select targets with PAL. A participant summarized his impressions by saying that the PAL provided a sense of “instant gratification” and “just felt right”. Nonetheless, some participants reported that the PAL's activation threshold was too sensitive, causing false activations. Almost all participants felt that the DAL's timeout (400ms in our studies) took control away from them and was delaying them from achieving their goal. However, one participant felt that the DAL felt “natural”. Finally, many participants were concerned that holding the stylus would be difficult if they were tired or if their grasp was perturbed by the environment – e.g., while standing or walking. Participants had mixed feelings toward the TL. Having the trailing lens permanently active was seen by some as appealing, but others found it frustrating that the (trailing) lens sometimes occluded a target. We observed a similar contrast of opinions regarding the TL's viscous behavior. While users described this behavior as “feeling

good”, they also found that at times it was difficult to catch the lens – i.e., participants would move the stylus slowly, thus pushing the lens away, or they would move the stylus too fast, thus overshooting the lens area. Finally, participants felt that having the lens area directly on top of the target (as it was the case with the PAL and the DAL) seemed less distracting than having the lens area at-a-distance (as was the case with the TL).

## EXPERIMENT 2: FREE-FORM ACTIVATION & ACCESS

Results from our first experiment suggest that there is no advantage in using our pointing lenses for targets larger than 4 pixels. While this is consistent with Ren and Moriya’s analysis of non-magnified target acquisition using stylus devices [17], our first experiment “forced” users to use the given lens for all selections, regardless of whether or not they felt they needed the expansion for selecting a particular target. It is unclear from the first experiment whether performance characteristics were affected by this forced lens usage. Thus, we conducted a second experiment, similar to the first, but where users were given the freedom to decide whether or not to use the available pointing lens to perform a selection. Thus, we are able to determine how often and for what target sizes users would choose to take advantage of pointing lenses.

### Participants, Apparatus, Task & Stimuli

Six volunteers, 18-44 years old participated in this study and were recruited from the University of Toronto population through e-mail lists. None had participated in experiment #1 and they did not receive compensation. We used the same apparatus, task and stimuli as in experiment #1.

### Procedure and Design

We used a 3 technique (*DAL*, *PAL*, *TL*) × 3 target size (2, 4, 8 pixels) within-participants design. Since the first experiment indicated that 4-pixel targets were the threshold at which performance of having no lenses was similar to using the lenses, we simply chose target sizes that “bracketed” this threshold, thus omitting the smallest 1-pixel target. For this experiment, participants were free to use or not use the available pointing lens technique for each selection. Thus, we recorded the lens activation frequency. Since our goal was to investigate lens usage and activation frequency, there was obviously no point in including the no lens condition. Other than these differences, the procedure and design for this experiment was identical to the previous one. In summary, the experiment consisted of:

6 participants × 3 techniques × 4 blocks × 3 sizes × 10 targets × 2 repetitions - 3 techniques × 10 warm-up selections = 4290 selection tasks.

### Results

The experiment took an average of 40 minutes per participant. We performed a 3 (technique) × 3 (target size) × 4 (block) RM-ANOVA on activations, on the logarithmically transformed selection times and on the number of errors. The presentation order of the techniques had no effects on activations, selection times or errors.

### Lens Activation

We observed no main effect for technique ( $F_{2,6}=0.435$ ,  $p=0.666$ ). Similarly, pairwise comparisons reveal no significant differences between techniques ( $p=1$ ). These results suggest that the participants’ need for a lens was not a function of the particular lens technique (Figure 14).

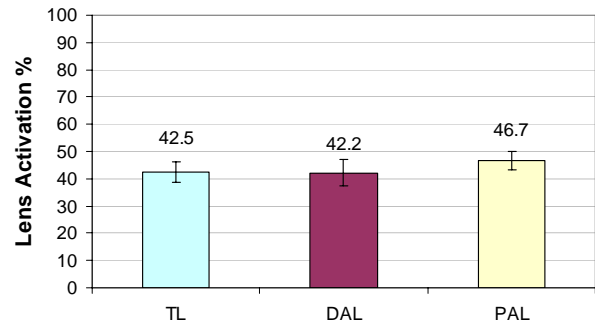


Figure 14: Activation % (+/- SE) per technique.

As we expected, there was a significant main effect for target size ( $F_{2,6}=193.16$ ,  $p<0.0001$ ). Pairwise comparisons show significant differences between all pairs ( $p<0.025$ ). Even though lenses were used for all target sizes, they were most used for 2-pixel targets (~90%) Lenses were used fairly often for 4-pixel targets (~35%) and rarely used for 8-pixel targets (~4%). Thus, lens’ activation frequency was a function of the target’s size. We did not observe technique × size interaction ( $F_{4,12}=1.393$ ,  $p=0.294$ ), and our post-hoc comparisons reveal no significant differences between techniques at different target sizes. That said, participants seemed to activate the PAL more often with 4 pixel targets than the DAL and the TL (Figure 15). We believe, based on observations during the experiment, that this difference is due to the PAL technique’s fluid nature that elicited participants to use it slightly more frequently.

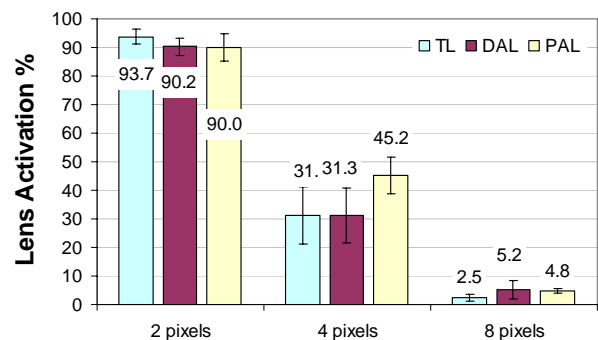


Figure 15: Activation % (+/- SE) per size and technique.

There was no effect for block ( $F_{3,9}=0.493$ ,  $p=0.696$ ) nor pairwise significant differences between blocks ( $p>0.45$ ). There was also no technique × block interaction ( $F_{6,18}=1.173$ ,  $p=0.363$ ) nor size × block interaction ( $F_{6,18}=1.285$ ,  $p=0.313$ ). This evidence suggests that the participants’ frequency of use of a lens for a particular target size remain the same regardless of the experimental block.

The results from this experiment confirm our hypothesis that pointing lenses are useful for targets up to 4 pixels. Furthermore, our data also hints that pointing lenses may still be useful for targets between 4 and 8 pixels in size.

### Selection Time

We observed no main effect for technique ( $F_{1,11}=110.391$ ,  $p=0.060$ ) or technique  $\times$  size ( $F_{1,11}=1.341$ ,  $p=0.453$ ) (Figure 16). It is interesting that in contrast with our first experiment there was no significant performance difference between techniques. We believe this is due to noise coming from cases where a lens was not used. If we separate the data by lens activation, we see the same patterns between techniques as in our first experiment: participants performed faster selections with the PAL for targets of size 2 and 4 when lenses were active ( $p<.0001$ ) (Figure 17).

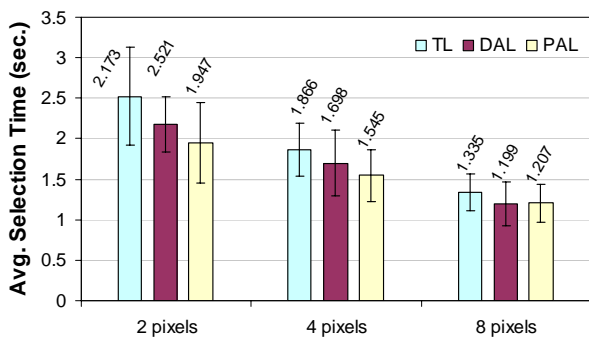


Figure 16: Average selection time (+/- SE) per size, technique.

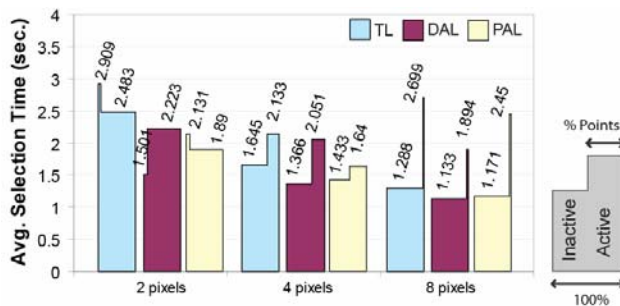


Figure 17: Avg. selection time per size, technique, activation.

As before, there was an effect for the *short* and *long* distance, but no distance  $\times$  technique interaction.

### Errors

There was no significant effect for technique ( $F_{2,6}=0.036$ ,  $p=0.446$ ) or technique  $\times$  size ( $F_{4,12}=0.52$ ,  $p=0.723$ ). Post-hoc comparisons also show no significant differences between techniques for targets of 2, 4 ( $p=1$ ) and 8 ( $p>0.068$ ) pixels. There was a significant effect for size ( $F_{2,6}=7.361$ ,  $p=0.024$ ), where participants committed the fewest selection errors for 8 pixel targets (Figure 18). It is interesting to highlight some observations when we separate the data between selections where participants did and did not activate a lens (Figure 19). For 2-pixel targets there seems to be no difference between missed selections done with and without a lens for the DAL and the PAL. However, in this case we

observe participants seemed to commit more errors (an average of 0.53 errors vs. an average of 0.20 errors) when not using the lens area of the TL.

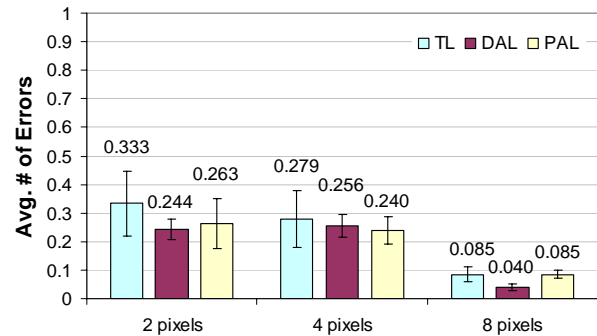


Figure 18: Average # of errors (+/- SE) per size and technique.

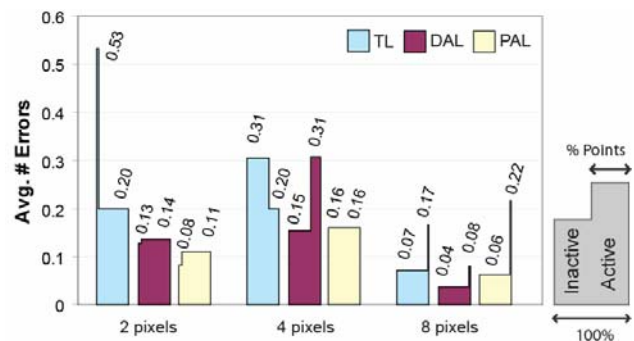


Figure 19: Average # of errors per size, technique, activation.

### Qualitative Feedback

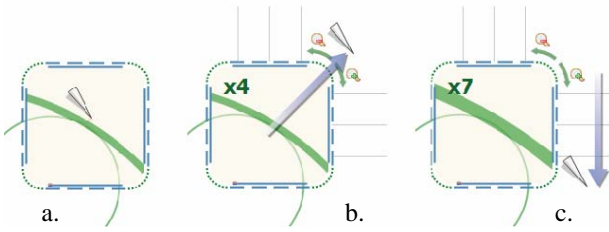
We presented participants with the same qualitative questions used in the first study, and their responses were similar. Again, when selecting targets, participants preferred the PAL (mean=5.42, SE=0.29) (agree / strongly agree) to the DAL (mean=3.85, SE=0.59) (neutral / somewhat agree) and the TL (mean=2.42, SE=0.42) (somewhat disagree / neutral). We also asked participants to rank the techniques in order of preference from 1 (best) to 4 (worst). The PAL ranked again highest (mean=1.14, SE=0.14) followed by the TL (mean=2.71, SE=0.18) and the DAL (mean=2.14, SE=0.26). In light of the results of our first study, it is interesting to observe how participants' preferences are divided between the DAL and the TL.

### DISCUSSION

The PAL comes out of our studies as the top performer both in terms of selection speed and users' preferences. It is also comparable with the DAL and the TL in terms of accuracy. Also, our results reveal not only that participants find pointing lenses useful for 4 pixel targets or smaller, but also suggest that this benefit may still occur for targets up to 8 pixels in size. Pointing lenses are thus important, since there are many scenarios in current GUIs where users need to select targets in this size range. Their value is further amplified when one interacts with imprecise input devices, or in adverse environments such as in a train or plane, which cause even larger targets to be difficult to select.

### Zoom Factor Adjustment

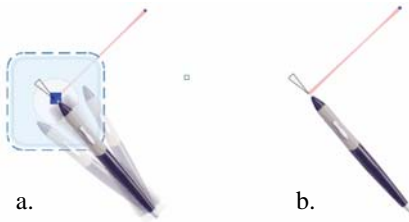
Our experiments use a fixed zoom factor of 4, but one can imagine that there are scenarios where users need to adjust that factor. Users could adjust a lens' zoom factor through a configuration dialog window, but such a method breaks the interaction's flow. We believe that it is better to have a more fluid mechanism for adjusting a lens' zoom factor. We propose a method of adjusting the zoom factor that is based on crossing widgets [2]. We define each of the 4 corners of a lens as a zoom-crossing segment. If one crosses a segment from the inside, a series of "zooming whiskers" are displayed on the segment's two neighboring edges (Figure 20). Each clockwise cross of a whisker will increase the zoom factor by one, while each counter-clockwise whisker cross will decrease the zoom factor by one. Re-entering the lens dismisses the zooming whiskers and allows users to either resume the current pointing task or re-enter the zoom-adjusting mode. All lens activations start at the last zoom factor selected.



**Figure 20: Detail of the zoom adjustment. (b) Stylus crosses a corner. (b-c) Zooming whiskers appear. (c) Crossing 3 whiskers clockwise increases the zoom by a factor of 3.**

### Rubbing Cross-Scale Dragging

Here we explore an additional interaction that enables cross-scale dragging with our lenses' designs. We can use a rubbing mechanism similar to the one described by Olwal and Feiner [13] to facilitate cross-scale dragging. One would rub with the stylus while dragging in the lens' area to switch to normal scale (Figure 21).



**Figure 21: (a) A rubbing gesture while dragging inside the lens' area switches the scale back to normal. (b) A dragging beam connects the stylus with the screen's cursor real position.**

A dragging beam bridges the separation between the stylus and the screen's cursor (Figure 21b). Users can perform a rubbing gesture while dragging in normal scale to bring back the lens' area, and continue with a more precise drag if needed.

### Pointing Lens Benefit Analysis

With a pointing lens, the action of selecting a target of size  $W$  at distance  $D$  consists of: a) pointing in the region of the target, b) activating the lens and c) selecting the target inside the lens. We can estimate the time part a) takes ( $MT_1$ ) using Fitts' law – i.e., for the target to be inside the lens once activated, the user must point to an area of size  $L/S$ , where  $L$  is the width of the lens and  $S$  is its zoom factor. For the purposes of this analysis we will consider the activation time of a lens as a constant  $MT_2$ , which depends on a lens' own activation method. Finally the time it takes to perform part c) can also be estimated using Fitts' law – i.e., the stylus is at a distance  $d \ll L$  from the target of size  $W \cdot S$ . The total movement time for a pointing lens selection  $MT_L$  can then be expressed as:

#### Equation 1

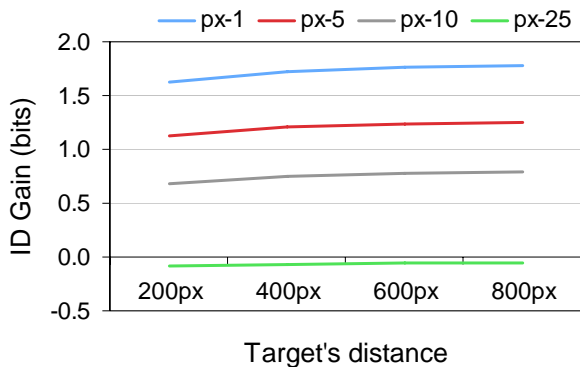
$$\begin{aligned}
 MT_L &= MT_1 + MT_2 + MT_3 \\
 &= a + b ID_1 + MT_2 + a + b ID_3 \\
 &= (2a + MT_2) + b(ID_1 + ID_3) \\
 &= A + b \left( \log \left( 1 + \frac{DS}{L} \right) + \log \left( 1 + \frac{d}{WS} \right) \right) \\
 &= A + b \log \left( \left( 1 + \frac{DS}{L} \right) \left( 1 + \frac{d}{WS} \right) \right) \\
 &= A + b ID_{lens}
 \end{aligned}$$

Figure 22 shows the gain  $ID_{gain} = ID_{no-lens} - ID_{lens}$  when using a lens of size 128. We use a value of  $d = L/4$ , a quarter of the lens' size. This is a conservative value based on our observations. The gain in ID is about 1.7 bits for 1-pixel targets, about 1.2 for 5-pixel ones, and just below zero for 25-pixel targets. It is interesting to note that this gain is not very sensitive to distance, as we found in experiment 1. Note also that the gain in ID does not translate completely into a gain in movement time because of the extra constant time for activating the lens (part b above) and for initiating the second pointing action. This explains why Figure 22 shows that lenses provide a gain in ID for targets up to 25 pixels wide while the experiments showed that the gain in time disappears between 4 and 8 pixels.

We can also use Equation 1 to explore what happens when  $S$  and  $L$  vary. If  $d$  is estimated as a fraction of  $L$ , then  $ID_{lens}$  does not change when  $L/S$  is kept constant. Our analysis reveals that increasing the zoom (or reducing the lens' size) maintains the advantage for small targets, but reduces it for larger ones.

This analysis is also useful in guiding the calibration of pointing lens parameters, such as the duration of DAL dwell, or the distance to the TL. For example, to gain a pointing advantage with 5-pixel widgets, Figure 22 shows that lens activation time cannot exceed a time cost equivalent to  $\sim 1.25$ bits. From experiment 1 we can obtain Fitts'

Law  $a$  and  $b$  parameters of  $-17\text{ms}$  and  $219\text{ms/bit}$ , suggesting that the dwell time should not exceed  $\sim 257\text{ms}$ .



**Figure 22: ID gain for lenses of size 128 and 4x zoom, not including the time to activate the lens.**

### CONCLUSIONS AND FUTURE WORK

Pointing lenses help users acquire and select targets on current GUIs by presenting an enlarged visual and interaction area. Much effort has been devoted to this topic in recent years. However, in the context of pen-computers, unique factors exist to make pointing challenging and no consistent solution has so far emerged or been incorporated. We presented and studied three pointing lenses for pen-based systems and found that the Pressure-Activated Lens is the top overall performer in terms of speed, accuracy and user preference. Nonetheless, we also found that for situations where pressure sensing is not available, both the Delay-Activated Lens and the Trailing Lens are adequate alternatives for users. Our experiment results not only show that our participants found pointing lenses beneficial for targets smaller than 5 pixels, but they also suggest that this benefit may extend to larger targets as well.

We have implemented both our proposed zooming and cross-scale dragging extensions in an interactive prototype. While preliminary user feedback on these extensions is promising, we believe that it is necessary to gather more quantitative and qualitative data regarding their use. Unlike other techniques pointing lenses performance is independent of target density. Also, pointing lenses implemented at the OS level will work with any existing application. We believe these key advantages of pointing lenses make them a significant contribution with broad implications,

Future work includes refining these lenses' activation techniques so as to reduce false activations, as well as revising those aspects of the lenses' design that can cause occlusion problems. Finally, we would like to extend our techniques to scenarios outside of pen-computers, such as interactive tabletops and touch-walls [3, 6, 16], which are capable of sensing simulated pressure input.

### REFERENCES

1. Albinsson, P.-A. and Zhai, S. (2003). High precision touch screen interaction. *CHI*, 105-112.
2. Apitz, G. and Guimbretière, F. (2004). CrossY: a crossing-based drawing application. *UIST*, 3-12.
3. Benko, H., Wilson, A.D. and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. *CHI*, 1263-1272.
4. Bier, E., Stone, M., Pier, K., Buxton, W. and DeRose, T. (1993). Toolglass and Magic Lenses: The see-through interface. *SIGGRAPH*, 73-80.
5. Blanch, R., Guiard, Y. and Beaudouin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *CHI*, 519-526.
6. Dietz, P. and Leigh, D. (2001). DiamondTouch: a multi-user touch technology. *UIST*, 219-226.
7. Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Exp.Psy.*, 47. 381-391.
8. Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B. and Kurtenbach, G. (2003). Tracking menus. *UIST*, 71-79.
9. Forlines, C., Balakrishnan, R., Beardsley, P., Baar, J.v. and Raskar, R. (2005). Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. *UIST*, 73-82.
10. Forlines, C., Vogel, D. and Balakrishnan, R. (2006). HybridPointing: Fluid switching between absolute and relative pointing with a direct input device. *UIST*, 211-220.
11. Grossman, T. and Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI*, 281-290.
12. McGuffin, M. and Balakrishnan, R. (2002). Acquisition of expanding targets. *CHI*, 57-64.
13. Olwal, A. and Feiner, S. (2003). Rubbing the Fisheye: Precise Touch-Screen Interaction with Gestures and Fisheye Views. *Conference Supplement of UIST*, 83-84.
14. Potter, R.L., Weldon, L.J. and Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *CHI*, 27-32.
15. Ramos, G. and Balakrishnan, R. (2005). Zliding: fluid zooming and sliding for high precision parameter manipulation. *UIST*, 143-152.
16. Rekimoto, J. (2002). SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *CHI*, 113-120.
17. Ren, X. and Moriya, S. (2000). Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *TOCHI*, 7 (3). 384-416.
18. Worden, A., Walker, N., Bharat, K. and Hudson, S. (1997). Making computers easier for older adults to use: area cursors and sticky icons. *CHI*, 266-271.

## **Contribution and Benefits Statement**

Presented three Pointing Lenses that improve stylus-based input. Our studies show that lenses are beneficial for targets smaller than 5 pixels, and that this benefit may extend to larger targets.

## **Pointing Lenses - Revisions:**

(pg .1) Changed Figure 1 to reflect a task people better relate to.

(pg.1) Introduction: We expanded the motivating discussion in our introduction to include more concrete examples in current GUIs where the acquisition of very small targets is commonplace (as we enumerate in our rebuttal).

(pg. 2 – 1<sup>st</sup> col, 2<sup>nd</sup> para.) Related Work: We added cases where previous solutions like bubble cursor do not scale and pointing lenses do.

(pg. 4 – 2<sup>nd</sup> col. 3<sup>rd</sup> para & pg. 7 1<sup>st</sup> col. 3<sup>rd</sup>. para.) Participants: Added additional information about our user population.

Addresses various other minor formatting and typographical issues and fixed equation 1, figures 16, 17 & 22