Increasing the Precision of Distant Pointing for Large High-Resolution Displays

Regis Kopper *Mara G. Silva* *Ryan P. McMahan* *Doug A. Bowman*
Center for Human Computer Interaction
Department of Computer Science
Virginia Tech
2202 Kraft Dr.
Blacksburg, VA 24060
Tel: 1-540-231-2058
{kopper, mara, rymcmaha, bowman}@vt.edu

ABSTRACT
Distant pointing at large displays allows rapid cursor movements, but can be problematic when high levels of precision are needed, due to natural hand tremor and tracking jitter. We present two ray-casting-based interaction techniques for large high-resolution displays – Absolute and Relative Mapping (ARM) Ray-casting and Zoom for Enhanced Large Display Acuity (ZELDA) – that address this precision problem. ZELDA enhances precision by providing a zoom window, which increases target sizes resulting in greater precision and visual acuity. ARM Ray-casting increases user control over the cursor position by allowing the user to activate and deactivate relative mapping as the need for precise manipulation arises. The results of an empirical study show that both approaches improve performance on high-precision tasks when compared to basic ray-casting. In realistic use, however, performance of the techniques is highly dependent on user strategy.

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

General terms: Large display interaction

Keywords: Ray-casting, high-precision

INTRODUCTION

Over the past few years, large high-resolution displays have become common in some application areas as the price of the technologies used to build this kind of display decreases. Some areas that benefit from such displays are geospatial visualization and visual analytics. Such large displays are not intended only for visualization, but also for interactive applications in which data can be manipulated across the entire display.

One of the attractions of large displays is the affordance to move freely in front of the display area. Interfaces based on mice and keyboards, however, discourage this type of movement, since these devices need to be parked on a flat surface to work properly, thus hindering the possibilities for the user to move around freely. Another disadvantage of the traditional mouse is the excessive clutching necessary to move the cursor between two distant areas of the screen.

Distant pointing is an alternative that enables the user to move around freely while still being able to interact with an application on a large display. One of the most common distant pointing techniques is *ray-casting* [1], in which the intersection of a ray extending from the input device with the screen determines the position of the cursor. One advantage of distant pointing over the mouse in large-display interaction is its speed to point at a desired region of the display. Since the cursor is absolutely positioned where the user is pointing, there is no delay due to rolling and clutching, as occurs with the mouse. If the large display does not have high resolution, or when high-precision pointing is not an issue, natural hand tremor will not interfere significantly with the interaction. For high-precision interaction on large high-resolution displays, though, hand and tracker jitter can hinder the usability of a distant pointing technique. The problem is amplified as users interact from a further distance, since the amplitude of the jitter will be amplified when the ray is projected onto the display.

We present two ray-casting-based high-precision interaction techniques for large high-resolution displays – Absolute and Relative Mapping (ARM) Ray-casting and Zoom for Enhanced Large Display Acuity (ZELDA) – that address this precision problem.

In the following sections we will describe related work; discuss the challenges inherent to distant pointing techniques; present ARM Ray-casting and ZELDA; describe our experiment; present results; and discuss the impact of both high-precision techniques on performance.

RELATED WORK

The HCI community has approached the precision problem from many angles. Approaches include increasing target size or target activation area [2, 3], zooming, adjusting mouse acceleration [4], and mapping a large display to a touchpad [5]. We make use of some of these ideas in the design of our techniques.

Distant pointing techniques also appear in the literature. Olsen and Nielsen [6] discussed interaction from a distance using laser pointer. Vogel and Balakrishnan [7] explored absolute ray-casting and relative pointing techniques for distant freehand pointing and clicking on large, high-resolution displays. Prior work has not fully addressed the precision problem with distant pointing, however.

DISTANT POINTING AT LARGE DISPLAYS

Pointing is one of the fundamental classes of 3D interaction techniques [1]. Within this class, ray-casting is a widely used technique that is simple but effective. With ray-casting, the user points with a virtual ray extending from the hand or input device. For large displays, the intersection point of this virtual ray with the display determines the location of interaction and often the cursor position.

Ray-casting with large displays provides users the freedom to move around in front of the display and supports moving the cursor rapidly to any point on the display. It is also very simple and direct. Unfortunately, its lack of precision makes it an impractical technique to use with large, high-resolution displays [7]. This is evident when trying to select and manipulate small targets.

Here we list issues that make ray-casting difficult to use with large displays for high-precision tasks. We are not the first to identify these issues, but we feel it is useful to gather all of them in one place:

Natural Hand Tremor. The hand has a physiological tremor around 8-12 Hz [8]. The tremor is of low amplitude, but when a user is using ray-casting at a distance, even a small tremor can cause the cursor to move many pixels.

Heisenberg Effect. When pressing a button on a device held in space, a user often slightly and unintentionally changes the position and orientation of the device. This is sometimes called the Heisenberg effect [9]. For ray-casting, the Heisenberg effect may cause the virtual ray to intersect unintentional locations when a click occurs. This is also an issue with freehand interaction [7].

Mapping Varies with Distance. With ray-casting, if a user is close to a display and rotates the ray by a certain angle, the motion mapped onto the display is much smaller than if the same rotation were performed from a greater distance to the display. This makes it extremely difficult to perform small motions when standing far away from the display.

No Parkability. With a mouse, a user can position the cursor in a desired position and then release the mouse, which will maintain the same position. This *parkability* [10] is due to the surface supporting the mouse. There is no parkability with ray-casting based on freehand pointing or handheld devices (although clutching can be used).

No Supporting Surface. A user using a mouse can also refine the position of the cursor with small movements due to the supporting surface. With ray-casting based on freehand pointing or handheld devices, small motions are harder to control because of the lack of a supporting surface.

Basic Enhancements to Ray-casting

Before developing our high-precision 3DI techniques, there were some basic enhancements that we applied to ray-casting to reduce problems like losing the cursor, tracking jitter, natural hand tremor, and the Heisenberg effect. We increased the cursor size (64 by 64 pixels) to allow users to see the cursor from greater distances. To eliminate the effects of tracking jitter and natural hand tremor, we used a dynamic recursive filter, as described by Vogel and Balakrishnan [7]. To reduce the Heisenberg effect, we implemented a technique that we refer to as *framing* (similar to the adjustments made by Vogel et al. [7]). Movement of the virtual ray while a click occurs is *framed* (ignored) and instead, the position of the ray at the beginning of the click is used.

HIGH-PRECISION DISTANT POINTING TECHNIQUES

To address the problems described above, we designed two high-precision distant pointing techniques.

Absolute and Relative Mapping (ARM) Ray-casting

Relative mapping of the cursor movement has been used as a solution to the problems of jitter and the Heisenberg effect in previous research [7], but there is still room for improvement in the transition between the absolute and relative mappings.

ARM Ray-casting utilizes bimanual input to provide an easy and smooth method for transitioning between absolute and relative mapping for ray-casting. The dominant hand is used for standard distant pointing with an absolute mapping, which affords fast coarse-grained interaction. The dominant hand is also used for standard input, such as left mouse button clicks.

The non-dominant hand is used to transition between mappings by holding down an input button to signal the use of a relative mapping, as illustrated by Figure 1. When the relative mapping button is first pressed, the current ray-casting intersection point is saved and considered as the relative mapping origin, R_O. As long as the relative mapping button remains pressed, further ray-casting intersections are then processed as vectors from the relative mapping origin, V_A, and scaled down by a scale factor, S, to produce shorter vectors, V_R. These shorter vectors effectively map standard ray-casting into a smaller defined area of interaction and provide higher levels of precision. The user perceives the relative mapping as a "slow motion" cursor, which appears to slow down by a factor of S when active. When the relative mapping button is released, the current position or cursor jumps back to the ray-casting intersection point for absolute mapping.

The scale factor, S, maps absolute pointing into a smaller relative area of interaction and offers increased precision. The value of S can be either application-specific or dynamically set by the user. For this research, we chose to set S to 0.1, which effectively maps absolute ray-casting to an interaction area that is a tenth of the total size of the display.

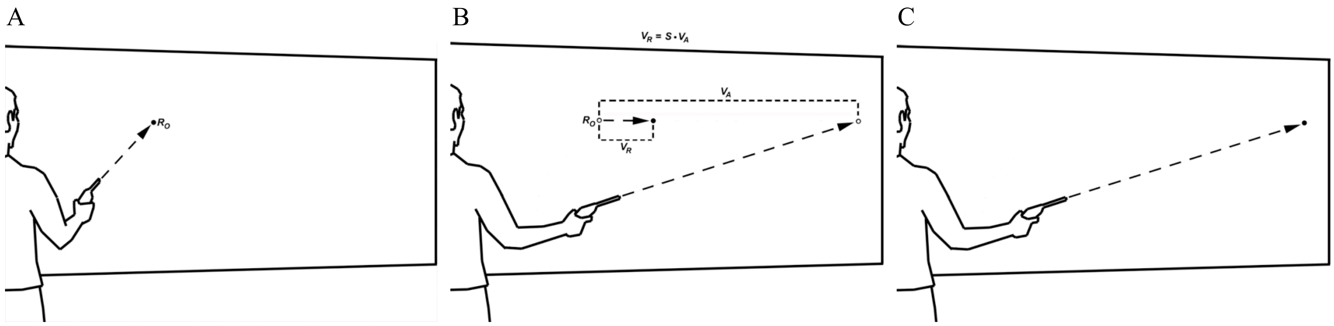

Figure 1: ARM Ray-casting. (A) The user activates relative mapping at the R_0 position. (B) The user points absolutely to the new position determined by the vector V_A but the cursor appears in the relative position determined by V_R, which is V_AS. (C) The user de-activates relative mapping, causing the cursor to return to the absolute mapping position.

In general, ARM Ray-Casting will be most effective when the user does not overuse the relative mode. If the relative mapping is activated too soon, the area where it is possible to move the cursor decreases by *S* and the cursor movement seems to slow down by the same amount. A good strategy when using ARM is to perform large, coarse movements using absolute pointing and only activate relative mapping when precision is really crucial.

ARM Ray-casting addresses the issue of precision for distant pointing in large high-resolution display settings. It does not, however, provide any difference in the user's visual perception of the objects in the application. When objects are very small or the user is very distant, it may be possible to select an object with ARM, but the user may not be able to tell when the cursor is touching the object. The technique that we describe in the following section addresses this and other issues.

Zoom for Enhanced Large Display Acuity (ZELDA)

Zoom for Enhanced Large Display Acuity (ZELDA) uses two distant pointing devices to control a zoom window in addition to the regular cursor. Figure 2 illustrates its functionality. The main idea behind ZELDA is to provide a magnified view of objects in the zoom window, which not only improves precision, but also enhances visual acuity.

As with ARM, the ZELDA user uses her dominant hand for standard ray-casting and for left-click functionality. The device in the non-dominant hand is used to control the zoom window's position, size and zoom factor.

The zoom window can be either moving or frozen and a button on the non-dominant hand device is used to switch modes. When the zoom window is moving, the user is able to place it with absolute ray-casting. So as to not lose contextual information, the zoom window is semi-transparent when moving, and contains an inner rectangle that shows the area that it will zoom to when the window is frozen (Figure 2b). A scroll wheel on the non-dominant hand device is used to change the zoom factor when the user is pointing toward the display (Figure 2c).

When the user points to the side, the scroll wheel can be used to resize the zoom window's horizontal dimension; the vertical dimension can be resized when the user points up or down. In either case, dark yellow lines are drawn on two sides of the zoom window to indicate that resizing is possible. The zoom window is always resized around its center. In our implementation, the zoom window starts as a square of 1536px on a side, with the zoom factor set to 2, so that a square area of 768px on a side is doubled in size when the zoom window is frozen.

By offering the zoomed-in view, ZELDA not only increases precision for selection and manipulation tasks, but also increases the visual acuity for the zoomed area. This feature is useful for several reasons. First, for very precise tasks, such as selecting very small objects, or for precise placement in very tight places, it may not be enough to increase the precision of movement if the objects are too small to see or the placement area is too tight to verify if a given object is completely within this area. Another advantage of zooming is that it can be used for tasks that involve visual perception as well as interaction. For example, in tasks in which the user wishes to select only icons with a particular label, if the label is too small to read, the zoom window can allow the user both to read the label and select the icon. Finally, ZELDA's zoom window can be used by itself for pure visualization applications that contain very dense and detailed information that cannot be seen at the original zoom level. One example is geospatial visualization, in which very detailed maps are shown on large high-resolution displays. ZELDA can be used to display areas of such maps in more detail.

There are many possible strategies to employ with ZELDA. The user should choose the size of the zoom window and the zoom factor depending on the application and task. For example, for tasks which require the user to select several small icons that are scattered across the display, a good strategy is to have a small zoom window with a high zoom factor and move the zoom window on top of the next object to be selected. On the other hand, in a placement task that requires the user to place several objects in a very tight space, a good strategy is to keep the zoom window on top of the target area with a high zoom factor. The user should also keep in mind that there is a tradeoff between the zoom factor and the ease of placing the zoom area over any given object. In other words, keeping a constant window size, the greater the zoom factor, the smaller the area that will be zoomed (the inner rectangle).

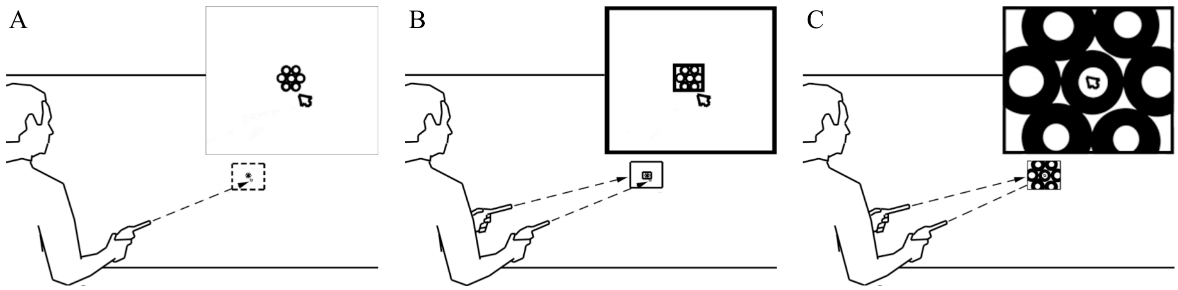

Figure 2: ZELDA technique. The insets show a detailed view of the cursor area. (A) The user just points at the display. (B) The user moves the zoom window over the objects so that the region of interest is within the inner rectangle. (C) The user freezes the zoom window and the area covered by the inner rectangle is magnified.

EXPERIMENT

We conducted an experiment to evaluate two aspects of the high-precision techniques we designed. First, we sought to verify that our techniques do indeed increase precision, compared to basic ray-casting, for interactive tasks on large high-resolution displays. For this purpose, we evaluated very simple tasks with automatic settings, which we call "atomic tasks." Second, we wished to evaluate the use of the techniques in more elaborate and realistic task settings. We refer to these as "complex tasks." For these tasks, users could use the techniques freely and were able to employ strategies that they judged most efficient. In both cases, basic ray-casting (with the filtering and framing enhancements) was used as a comparison technique.

Design

We used a mixed design in our experiment. Subjects were divided into two groups, one of which used ZELDA and the other used ARM Ray-casting. Subjects in both groups used basic ray-casting as a comparison.

Our hypothesis was that ARM and ZELDA are indeed more precise than basic ray-casting. The atomic tasks were meant to evaluate whether the techniques are more precise when the effects of user strategy are removed, so the system provided the strategy for the users. In the complex tasks, we wanted to determine whether the techniques afforded strategies that would lead to better performance.

Atomic Tasks. A trial consisted of clicking on a target, selecting an icon and dragging the icon completely into the target. These three actions define two subtasks: the *selection* subtask and the *placement* subtask. We analyzed each subtask independently.

The independent variables for the selection subtask were **icon radius** (I: 64px, 80px and 104px), **amplitude of movement** (A: 1500px and 4000px), **distance from the user to the display** (D: 125cm and 250cm) and **technique** (T). For the placement subtask, instead of using icon radius as the independent variable, we used **effective target radius** (R), which considers the available space in which an icon can move while being completely inside the target and is defined by the difference between the target and the icon radii. The target radius was always 128px, so the effective target radius was 64, 48 or 22px depending on the icon. The other independent variables were the same as the selection subtask.

One of the technique conditions was *basic ray-casting*, for both groups. For the group that used ZELDA, the other technique conditions were *ZELDA with a zoom factor of 2*, and *ZELDA with a zoom factor of 4*. In both ZELDA technique conditions, the zoom window was a square of 1536px on a side, and both the icon and target were always completely shown in the zoom window. For the ARM Ray-casting group, the technique conditions, in addition to basic ray-casting, were *threshold radius of 256px*, and *threshold radius of 512px*. The threshold was the distance from the center of the icon or target at which the relative mapping would be automatically activated. By controlling the zoom factor and the threshold size, we simulated different strategies that could be used but eliminated variance in user strategies. Of course, this design also removes the need for the user to position, zoom, and resize the zoom window. We chose this approach because we only wanted to verify the assumptions that slowing down the cursor or zooming in a region of interest do indeed increase precision.

The most important dependent variable for the atomic tasks was the **number of errors**, since that is directly related to precision. We also recorded the **time** for the selection subtask, which started by clicking on the target and ended when the icon was selected. Due to a malfunction in the time recording software, however, the time for the placement subtask was not recorded.

The study design was 3 (I) x 2 (A) x 2 (D) x 3 (T) within subjects, with T varying between subjects for ZELDA and ARM Ray-casting conditions. Thus, there were a total of 36 conditions for each subject. We took five measures per subject in each condition, and used the average for the analysis. We randomized the order of the conditions for I and A, and counterbalanced for D and T, so that users would perform all the conditions for a certain technique at a certain distance together.

Complex Tasks. The between-subjects independent variable for the complex tasks was the **technique** (ARM Ray-casting or ZELDA, along with basic ray-casting) and the order of execution was counterbalanced. Each technique condition contained six tasks: three of the selection type and three of the placement type (described below). The order of the task types was also counterbalanced and there was an easy, a moderate and a difficult task for each type. The order of difficulty was always from easy to difficult, so the subject had a chance to develop a strategy before performing the difficult task. The dependent variables were **time to complete the task** and **strategies** used.

Apparatus

We used a flat tiled display consisting of 50 NEC MultiSync LCD2080UXi monitors in a 10x5 configuration (Figure 3). Each monitor's resolution was 1600x1200 pixels, resulting in a total resolution of 96 Megapixels.

A wireless mouse with reflective markers was used as the primary input device, which we call the *cursor controller* (Figure 4 Right). We chose a mouse with a gun-like form factor, which affords a firmer grip than most wireless mice, thus reducing the Heisenberg effect when clicking. A virtual ray extending from the cursor controller determines the positioning of the application's cursor. We also used a wireless air mouse with reflective markers, which we call the *secondary controller* (Figure 4 Left).

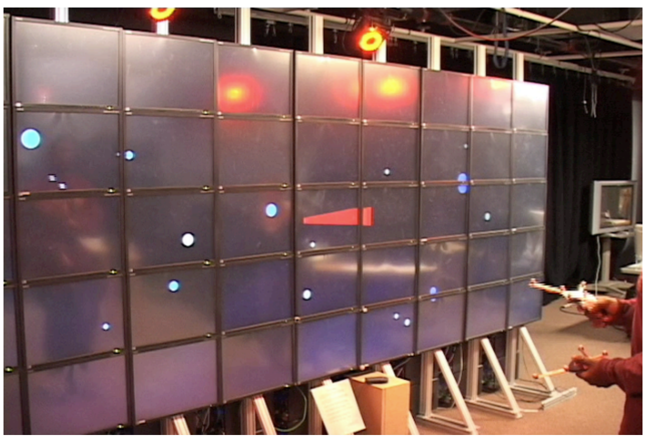

Figure 3: Large high-resolution display used in our experiment.

To enable six-degree-of-freedom input, we attached reflective markers to the wireless mice, which were tracked by a VICON MX system with eight cameras. A dynamic recursive low pass filter [7] was applied to the raw position data read from the tracker, which visibly reduced jitter without compromising the response time.

The application was implemented using the OpenScenegraph (www.openscenegraph.org) library. It consisted of icons and targets, whose positions were determined by an XML description file. The application worked in real time with no noticeable lag.

Subjects

Sixteen subjects were recruited from the university community and averaged 23.2 years of age with a standard deviation of 5.2. 13 subjects were male, and 3 were female. 12 participants were undergraduate, and 4 were graduate students. All subjects were either computer science or engineering majors. All participants in the study were right-handed and color-sighted.

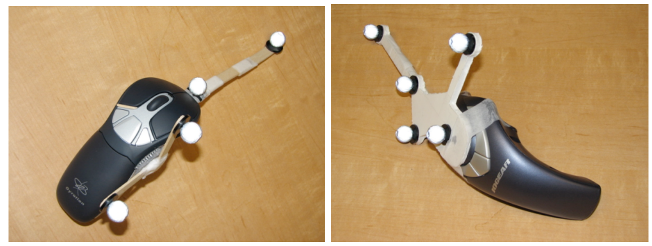

Figure 4: Figure 3: (Left) *Secondary controller:* a Gyration GO 2.4 Optical Air Mouse. (Right) *Cursor controller:* an logear Phaser Mouse.

Procedure

Subjects were first greeted by the experimenter and given a standard color blindness test. Next, they filled in a background questionnaire where they provided some demographic information as well as any previous experiences with large displays.

Atomic Tasks. To automate the use of the techniques in the ZELDA conditions, the zoom window was placed over the area where the subjects needed to perform the icon selection or placement. For the ARM conditions, we displayed a grey circle at the threshold distance. Once inside the threshold, the relative mode would automatically engage with a scale factor of 0.1.The zoom window and threshold automatically moved to the target area once the icon was selected.

Figure 5 shows the setup for one of the atomic tasks. The display contained two circular objects, representing a target on the right and an icon on the left. The trial started when the subject clicked on the target, which changed color from red to green. Next, the subject was instructed to select the icon by clicking with the trigger button on the cursor controller, and to drag it completely inside the target, at which point the trial finished and moved automatically to the next.

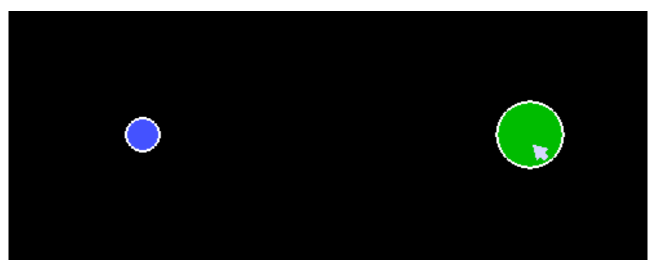

Figure 5: An atomic task setup.

Objects in these trials were always centered on the display, directly in front of the subject, who stood behind a line marked on the floor to mark the correct distance from the display.

Before the set of trials for each technique, subjects were given a guided practice so they could learn how to use the cursor controller and how to complete a task successfully. In the training session, subjects saw all the conditions once.

When the atomic tasks session was finished, subjects took a three-minute break and video recording was started for the complex tasks.

Complex tasks. In this session, the subjects were able to use the techniques freely, with all their features. Both the ZELDA and ARM Ray-casting groups performed the same tasks using basic ray-casting and the respective high-precision technique. The order of execution was counter-balanced among the users of each group.

The interaction metaphor for the complex tasks was similar to a desktop metaphor. Single icons could be selected via pointing and clicking, and multiple icons could be selected with a rubberbanded selection box. Any icon within a selection group could be clicked on to drag the selection. Clicking anywhere else cancelled the selection. The trigger button in the cursor controller behaved similar to the left mouse button in the desktop and could be used both for selecting icons and drawing the selection box.

Subjects followed a guided tutorial to learn basic ray-casting and then the respective high-precision technique. They were repeatedly reminded during the tutorial that they could move freely in front of the display, and that they did not need to stay at predefined position. We also asked the subjects to think about different strategies and to tell the experimenter how they would accomplish the task in an efficient way. Subjects practiced the techniques until the experimenter felt that they understood them completely.

In the experimental session, users performed two types of tasks, selection and placement, which each had three levels of difficulty. The selection tasks (Figure 6) consisted of selecting and dragging blue icons that were surrounded by gray ones that were not allowed to be moved. The task was completed when all blue icons were dragged inside a large target area. The placement tasks (Figure 7) required subjects to drag blue icons of various sizes into a trapezoid-shaped target. The icons had to be placed in order of size, so that all of them were completely inside the target without overlapping. Before completing the tasks, subjects were instructed to think about their strategy and only then click on the red target, which turned to green to indicate that the task had begun.

The difficult tasks required very high precision. In the selection type, all the blue icons were very small and closely surrounded by gray icons of the same size. The difficult placement task contained a very tight target, so that placing all the icons inside it left only a few pixels of target area to spare.

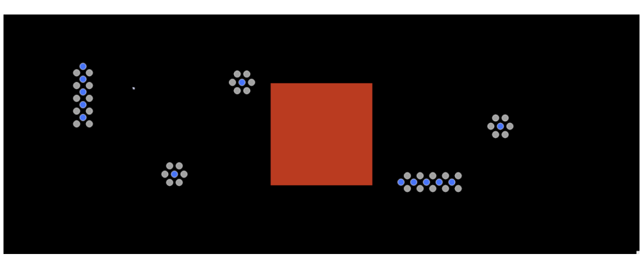

Figure 6: Complex selection task.

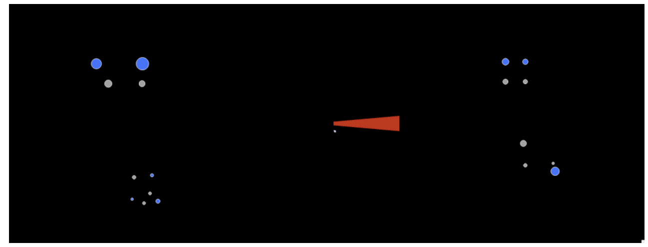

Figure 7: Complex placement task.

RESULTS

Atomic Tasks

We performed a full factorial analysis of variance (ANOVA) to compare the high-precision techniques we designed with basic ray-casting. We also compared ZELDA and ARM Ray-casting between subjects. Figure 8 shows the overall mean number of errors for the selection and placement subtasks in each technique condition. It is clear that ZELDA and ARM resulted in higher precision than basic ray-casting based on this metric.

ARM vs. Basic Ray-casting. An analysis of the time to complete the selection subtask showed that both ARM Ray-casting treatments (small and large threshold) were significantly slower than basic ray-casting ($F(35, 252)=10.96$, $p<0.0001$). This can be explained by the fact that the absolute amplitude of the movement increased greatly for the ARM Ray-casting conditions, since the relative mode would be automatically activated before the user reached the icon to be selected.

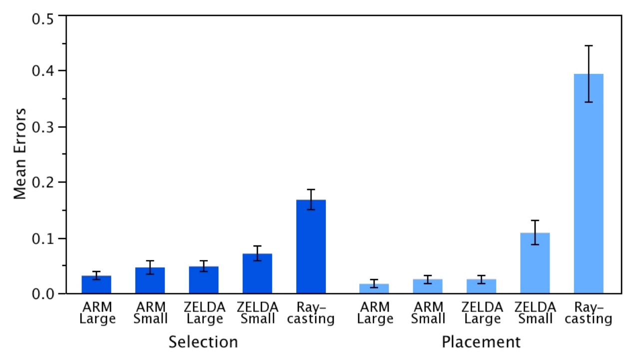

Figure 8: Mean number of errors for the selection and placement subtasks.

With respect to mean number of errors per trial, we found a significant main effect of technique for the selection ($F(35,252)=18.33$, $p<0.0001$) and placement ($F(35,252)=76.18$, $p<0.0001$) subtasks. Based on a post-hoc Tukey HSD test, the two ARM conditions resulted in significantly fewer errors than basic ray-casting in both subtasks.

In the selection subtask, there were significant interactions of technique with distance ($F(35,252)=6.98$, $p<0.0002$) and with icon radius ($F(35,252)=2.46$, $p<0.05$). Looking at the post-hoc Tukey HSD tests, we found that, for the technique vs. distance interaction, basic ray-casting was only significantly different than the other conditions when the subject was far from the display. This means that ARM Ray casting offers the greatest gains in precision when the user is far from the display. In terms of the interaction between technique and icon radius, the only conditions that were significantly different than any ARM Ray-casting conditions were with basic ray-casting with small icons. Again, this indicates that ARM Ray-casting is more useful when selecting small objects.

Since the conditions required more precision for the placement subtask, we found a higher number of significant interactions in this subtask. Interactions with the technique factor were distance ($F(35,252)=29.13$, $p<0.0001$) and effective target size ($F(35,252)=36.78$, $p<0.0001$). We also found a three-way interaction between technique, distance and effective target size ($F(35,252)=13.78$, $p<0.0001$). The post-hoc Tukey HSD tests with each interaction show that basic ray-casting when the subject was far from the display and/or interacting with the smallest icons were the only conditions significantly different than all the other treatments.

All these results support our hypothesis that ARM Ray-casting does indeed increase precision of distant pointing, especially when the target sizes are small and the user is distant from the display.

ZELDA vs. Basic Ray-casting. The two versions of ZELDA (with zoom factors of 2 and 4) were compared to basic ray-casting with a full factorial ANOVA for time and mean number of errors. The time analysis of the selection subtask shows a significant main effect of technique ($F(35,252)=33.13$, $p<0.0001$). A post-hoc Tukey HSD test shows that ZELDA with the large zoom factor was the significantly fastest condition, while basic ray-casting was the significantly slowest.

The analysis of the mean number of errors per trial for the selection subtask showed a significant main effect of the technique factor ($F(35,252)=16.74$, $p<0.0001$). The post-hoc test indicated that basic ray-casting had significantly more errors than both ZELDA treatments. We also found a significant interaction between technique and distance to the display ($F(35,252)=6.57$, $p<0.002$). A Tukey HSD post-hoc test showed that the only significantly different condition was basic ray-casting at the far distance. This supports the assertion that ZELDA is most useful for tasks that require a higher level of precision.

For placement subtasks, a main effect was found for technique ($F(35, 252)=28.02$, $p<0.0001$) and the post-hoc test showed that basic ray-casting produced a significantly higher number of errors than the ZELDA treatments. The observed interactions with technique were distance ($F(35,252)=12.58$, $p<0.0001$), effective target size ($F(35, 252)=10.46$, $p<0.0001$) and a three-way interaction with distance and effective target size ($F(35,252)=6.51$, $p< 0.0001$). The post-hoc Tukey HSD tests for each of the interactions showed that basic ray-casting when far from the display and/or with the smallest effective target size were significantly different than all the other conditions.

All the findings from the comparative analysis of ZELDA and basic ray-casting provide evidence that ZELDA indeed increases precision and is most helpful for the hardest tasks.

ARM vs. ZELDA. We performed a between-subjects full factorial ANOVA to compare the ZELDA and ARM techniques. For the response time in the selection subtask, there was a main effect of the technique, indicating that the ARM techniques were significantly slower than the ZELDA techniques ($F(51,332)=92.56$, $p<0.0001$). We believe that this behavior is due to the threshold, which increased the absolute amplitude of the movement of the ARM techniques.

The error analysis of the selection subtask did not show a significant main effect of technique ($F(51,332)= 2.23$, $p=0.090$). This suggests that all four high-precision techniques provided enough precision for the selection subtask. For the placement subtask, there was a main effect of technique ($F(51,332)=15.97$, $p<0.0001$), and the post-hoc test showed that ZELDA with the small zoom factor was significantly worse than all other techniques. With a zoom factor of 2, the smaller effective target sizes were still difficult to place icons into, since they only doubled in size (e.g., from 22 to 44px, which is still quite small). There were interactions of the technique with distance, technique with effective target size, and a three-way interaction between all these factors. In all cases, ZELDA with the small zoom factor, combined with the far distance from the display and/or with the small effective target size were the only significantly different conditions.

Complex Tasks

Time Analysis. Time to complete complex tasks varied according to the difficulty and nature of the task (selection or placement). The average time for selection tasks ranged from 23s to 59s, while for placement tasks the averages were between 54s and 150s.

We performed a one-way ANOVA to compare basic ray-casting with ZELDA and ARM. As we expected, the variance was too large to result in statistically significant differences, since user strategy played an important role in task performance. The only exception was a significant effect of technique for the difficult selection task, showing that ZELDA was significantly slower than basic ray-casting in a within-subjects analysis ($F(1,14)=4.85$, $p<0.05$). For the same task, ZELDA was also significantly slower than ARM in a between-subjects analysis ($F(3,12)=6.49$, $p<0.05$). Figure 9 illustrates these differences. For this task in particular, participants could achieve high precision by moving closer to the display and still complete it fairly quickly. For such a quick task, the time spent setting up the zoom window for ZELDA had a significant impact on the overall time. Half of the participants who used ZELDA spent time resizing the zoom window for this task and the other half did not have a large enough window to fit the array of icons to be selected.

The mean times for completion of the difficult placement tasks are also shown in Figure 9. Even though there are no statistically significant differences among these times, we found that four out of eight participants improved their performance by 15% or more using ZELDA as compared to basic ray-casting, and four out of eight participants had this level of improvement using ARM Ray-casting. Since performance is so dependent on strategy, it is relevant to analyze the strategies that caused some users to improve or decrease their performance, and to evaluate how well our techniques supported user preferences.

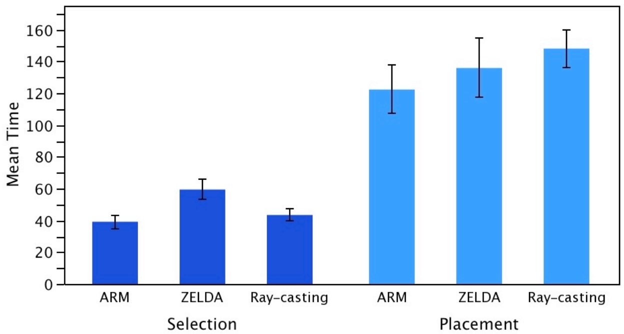

Figure 9: Mean time for completion of difficult selection and difficult placement tasks.

Analysis based on user strategies. We analyzed user strategy by reviewing the video recordings of each subject's complex task performance. A user strategy encompassed a set of decisions ranging from physical navigation in front of the display to how they chose to use their assigned technique. Participants chose strategies based on their skills and personal preferences. It is important to mention that some users changed their strategy from one task to another, and even during the completion of a task.

Among physical navigation strategies employed by users, the most common were moving around in the space close to the screen to interact closely with the display, standing in a distant fixed position during the completion of a task, moving only short distances around a particular point, and moving only if absolutely necessary to complete of the task.

An interesting finding was that most participants prefer to maintain some distance from the display. One participant even mentioned that he felt uncomfortable working very close to the screen. In our study only five out of the sixteen subjects consistently used the area up close to the display. Most users worked primarily in an area from 125cm to 250cm away from the display, only moving closer if necessary, as was the case for the difficult tasks using basic ray-casting. When using basic ray-casting, these users had to choose between better performance (stepping forward to work up close), and personal preferences (remaining at a comfortable distance). Compared to basic ray-casting, eleven participants interacted at a greater distance from the display when using ARM or ZELDA. These techniques enabled them to interact with the large display from a comfortable distance without a major impact on their performance.

A related insight is that most participants preferred to minimize their walking, which means that, given the option, they preferred not to move. This finding was supported by the comments of some participants, who stated that the zoom window in ZELDA eliminated their need to walk around during tasks. Ten participants performed less physical navigation when using one of the high-precision techniques, compared to basic ray-casting. One participant did not move at all for both basic ray-casting and his assigned technique.

It is of paramount importance to support physical navigation strategies, since users will use a technique or technology in a way that is comfortable to them. We have seen that our high-precision techniques improve (or at least maintain) performance for users applying all categories of physical navigation strategies, so we conclude that both ARM Ray-casting and ZELDA enable users to achieve high precision while supporting their personal choices.

When using ZELDA, users had multiple settings that allowed them to customize the zoom window in terms of zoom factor, width, height, and placement. Setting up the zoom window takes time, and users had to consider that in their strategies. Some users preferred to use the zoom window only for the most difficult tasks, while others used it for almost all interaction with icons, no matter the level of difficulty. During the placement tasks, some users would place the zoom window over the target, and change its size to encompass the whole target, thus making it very large. Even though it took them extra time to set up the window, the placement of icons inside the target then became an easy task. On the other hand, some users preferred not to spend time changing the zoom window settings, so they placed and used it as-is over the area where they were interacting at the moment, and then move the zoom window wherever it was needed next.

Results for the difficult placement task showed that ZELDA helped four out of eight participants to improve their performance by 15% or more compared to basic ray-casting. Three other participants took a similar time between conditions, and one participant had worse performance with ZELDA. By comparing the similarities and differences of their strategies we gained insights about good and poor choices of strategies.

We noticed that participants with improved performance chose a strategy at the beginning of the task, and used it consistently throughout the task. Most of them resized the zoom window to make it very large, placed it over the target, and changed the zoom factor, resulting in an immense target. Since the window was stationary users had to put more effort in selecting small icons around the display, but this effort was compensated by the ease of aligning the icons inside the target. Another strategy was to always move the zoom window over the icon with which the user is currently interacting, without changing any setting on the window (size and zoom factor). Even though these users had to move the window frequently, that was compensated by the fact that they did not spend time adjusting settings.

One of the poor strategies chosen by participants who did not improve their performance was changing the strategy during the course of the task: i.e., going back and forth between using the zoom window as-is over the area of the current interaction and resizing it to fit the whole target. Another poor strategy was only using the zoom window for the selection of small icons, but not during the alignment of the icons inside the target, which was the most difficult part of the task.

Strategies for the use of ARM Ray-casting varied only in terms of when to activate the relative mapping. Due to the directness of controlling it, participants used the relative mapping for almost all tasks, helping them to select icons, to place icons, and to start and finish selection boxes. We observed that participants enjoyed the fine control over the cursor movement, making comments like "I enjoyed the precision tool to slow the mouse movement down". We noticed that most users would activate relative mapping even for easy tasks. The extra time added by the slow movement of the cursor was compensated by the confidence that the users gained about clicking at the right place on the first try, avoiding errors or multiple attempts.

We compared time results of the eight participants that used basic ray-casting and ARM Ray-casting to complete the difficult placement task: four subjects improved their performance by 15% or more; two performed similarly; and two had worse performance with ARM. After analyzing the strategies of all subjects in this group, we found that the four users who did not improve performance with ARM Ray-casting were special cases. One user mistakenly arranged the icons in the wrong order inside the target and had to rearrange them, greatly increasing the completion time. Interestingly, the other three participants who did not improve using ARM were actually extremely good with basic ray-casting. These three participants were among the four best basic ray-casting times for the difficult placement task. They learned how to achieve high precision using basic ray-casting, and the slow movement of the cursor in ARM Ray-casting actually slowed them down. They still performed well using ARM, which can be seen by the fact that all three completed the task faster than the average ARM Ray-casting time.

DISCUSSION

The techniques we designed, ARM Ray-casting and ZELDA, offer the possibility to interact precisely with objects that are difficult to manipulate using basic ray-casting, as we showed in the analysis of the atomic tasks. In real-world tasks, however, employing a good strategy is as important as using the high-precision technique. As we found in the complex trials, when objects are large or do not need to be precisely placed, basic ray-casting can be sufficient, and using too many of the features of ZELDA or ARM Ray-casting may simply add unnecessary overhead and complexity.

ARM Ray-casting can provide high precision without a great deal of additional overhead. One of its initial drawbacks is that it requires bimanual interaction, since the non-dominant hand controls the mapping mode while the dominant hand performs common tasks. We observed that users can overcome any coordination problems with few minutes of practice, making the technique quite lightweight even for inexperienced users. We learned from our experiment that ARM Ray-casting could be used for virtually any selection or manipulation task without hurting performance. The user just needs to know precisely when to activate and deactivate the relative mapping – only when high-precision pointing is needed. That may be to select a single object, to create selection boxes or to precisely place one or more objects in a region of the display. ARM could be improved if the technique itself could infer (perhaps based on movement speed) when the user is trying to interact with higher precision and activate the relative mapping automatically.

ZELDA is a more versatile technique, but it comes at the cost of complexity. We observed in our study that a great amount of time may be used setting up the zoom window. The users who performed better with ZELDA were the ones who minimized the number of zoom window operations, be it by fixing the zoom window at a specific size and shape that would be most beneficial at the beginning of a task, or by performing as few zooming and resizing operations with it as possible, thus moving the zoom window more frequently.

The distant pointing techniques that we designed both serve the purpose of increasing precision, but they achieve this goal through different means. They are actually complementary in many ways, and could be combined to offer more power and flexibility. The naïve way to combine ZELDA with ARM Ray-casting would be to add another button to the secondary controller, so that one button would control the operations with the zoom window whereas the other would control the ARM Ray-casting mapping mode. Such a combined technique would provide all the features of both ZELDA and ARM Ray-casting techniques, but with even more overhead due to complexity and coordination.

But there are other ways to combine the good features of ZELDA and ARM Ray-casting; for example, a zoom window could pop-up around the cursor area every time relative mode is activated, thus eliminating the need to set it up to a specific position. This would keep the technique as simple as ARM Ray-casting, while adding the enhanced visual acuity of ZELDA.

One of the interesting findings from our study is that most users prefer to stay at a certain distance from the display while interacting and only move up close when absolutely necessary. This seems to contradict Ball's findings [11] which showed that users prefer to walk rather than using virtual navigation for large display tasks. We believe that this difference is due to the nature of the tasks in each experiment. While we investigated interactive tasks that did not require physical navigation or detailed views, Ball's tasks were essentially search tasks that involved small features in the display and required navigation, either physical or virtual, to be completed. Our finding is interesting because it indicates that people manipulating objects on large displays would rather keep an overview than moving close to the display. This has important implications for future large display interface design.

Overall, it seems that ZELDA and ARM Ray-casting allow users to be "lazy" both in terms of pointing accuracy and physical navigation, while maintaining precision and efficiency in most cases. Basic Ray-casting can achieve good precision and efficiency, but forces users to work harder.

CONCLUSIONS AND FUTURE WORK

We have presented and evaluated two high-precision interaction techniques based on distant pointing for large, high-resolution displays. We demonstrated that these techniques improve the precision of selection and manipulation of objects compared to basic ray-casting when strategy is not involved. We also analyzed subjects' performance of complex tasks that required high precision and strategy. Depending on the strategy, users may benefit more or less from the high-precision techniques. Users of our techniques could use strategies that allowed them to stand at a greater distance from the display, which was a common preference among the subjects in our study.

Our study has also shown that it is feasible to use distant pointing (3D) interaction even with 2D data, and that it is indeed possible to increase precision. With more realistic tasks, performance depends heavily on strategy and we found strategies that work well with each technique.

For future work, we plan to analyze how our techniques can be combined together in order to offer better high-precision interaction for large high-resolution displays. It is also important to compare the 3D interaction techniques that we created with 2D techniques that have been proposed for large display interaction.

Finally, we believe that there are models of human motor behavior for distant pointing that allow for the prediction of performance according to different strategies that can be taken to solve a task. We are studying such models that could offer guidelines for effective distant pointing interaction for large high-resolution displays.

REFERENCES

[1] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[2] T. Grossman and R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," in *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2005, pp. 281--290.

[3] M. McGuffin and R. Balakrishnan, "Acquisition of expanding targets," in *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2002, pp. 57--64.

[4] A. Cockburn and A. Firth, "Improving the Acquisition of Small Targets," in *People and Computers XVII: British Computer Society Conference on Human Computer Interaction.*, 2003, pp. 181-196.

[5] S. Malik, A. Ranjan, and R. Balakrishnan, "Interacting with large displays from a distance with vision-tracked multi-finger gestural input," in *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, New York, NY, USA, 2005, pp. 43--52.

[6] J. Dan R. Olsen and T. Nielsen, "Laser pointer interaction," in *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2001, pp. 17--22.

[7] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, New York, NY, USA, 2005, pp. 33--42.

[8] C. Riviere, R. S. Rader, and P. Khosla, "Characteristics of hand motion of eye surgeons," in *19th Annual Conference of the IEEE Engineering in Medicine and Biology Society*, 1997, pp. 1690 - 1693.

[9] D. A. Bowman, C. A. Wingrave, V. Q. Ly, and C. J. Rhoton, "Novel uses of pinch gloves for virtual reality environment techniques," in *Virtual Reality*, 2002, pp. 122-129.

[10] R. Balakrishnan, G. W. Fitzmaurice, and G. Kurtenbach, "User interfaces for volumetric displays," *Computer,* vol. 34, pp. 37-45, Mar 2001.

[11] R. Ball and C. North, "Effects of tiled high-resolution display on basic visualization and navigation tasks," in *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, New York, NY, USA, 2005, pp. 1196--1199.