

Research Article

Large Display Interaction via Multiple Acceleration Curves and Multifinger Pointer Control

Andrey Esakia, Alex Endert, and Chris North

Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA

Correspondence should be addressed to Andrey Esakia; esakia@vt.edu

Received 14 April 2014; Accepted 14 October 2014; Published 25 November 2014

Academic Editor: Sebastian Moeller

Copyright © 2014 Andrey Esakia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Large high-resolution displays combine high pixel density with ample physical dimensions. The combination of these factors creates a multiscale workspace where interactive targeting of on-screen objects requires both high speed for distant targets and high accuracy for small targets. Modern operating systems support implicit dynamic control-display gain adjustment (i.e., a pointer acceleration curve) that helps to maintain both speed and accuracy. However, large high-resolution displays require a broader range of control-display gains than a single acceleration curve can useably enable. Some interaction techniques attempt to solve the problem by utilizing multiple explicit modes of interaction, where different modes provide different levels of pointer precision. Here, we investigate the alternative hypothesis of using a *single mode* of interaction for continuous pointing that enables both (1) standard *implicit* granularity control via an acceleration curve and (2) *explicit* switching between multiple acceleration curves in an efficient and dynamic way. We evaluate a sample solution that augments standard touchpad accelerated pointer manipulation with multitouch capability, where the choice of acceleration curve dynamically changes depending on the number of fingers in contact with the touchpad. Specifically, users can dynamically switch among three different acceleration curves by using one, two, or three fingers on the touchpad.

1. Introduction

With large high-resolution displays, users are expected to perform pointer movements over a wide range of distances and target sizes, ranging from fast long distances to precise small targets. Such multiscale displays are capable of displaying small objects far apart from each other [1]. Fast and accurate acquisition of such objects imposes two main requirements on the pointing device: fast pointer movement for long distances [2] and high precision for fine-grained refinement of pointer position.

Control-display (CD) gain specifies the multiplier between the speed with which the user moves the control device and the resulting speed of the pointer on the display. Casiez et al. [3] identify the usable range of fixed CD gain values, depending on targeting characteristics. Gain must be large enough to enable movement across the entire screen without clutching yet small enough to enable precise pointing to small targets. Unfortunately, for large high-resolution displays the usable range is empty, causing fixed CD gain

to perform poorly on large displays [4], thus necessitating variable CD gain techniques such as pointer acceleration.

Multiple studies show [3, 5, 6] that automatic dynamic CD gain adjustment (i.e., pointer acceleration) improves pointing performance by dynamically adjusting CD gain depending on the physical movement speed. As a result, lower speeds decrease CD gain for precision, while higher speeds increase the gain for faster pointer movements. This provides users with *implicit* granularity control for relative pointing devices and defines a pointer acceleration curve that relates control speed to display speed (see Figure 1, e.g., pointer acceleration curves).

However, it seems unlikely that there exists a single acceleration curve that performs well over the diverse range of tasks on large displays. Casiez and Roussel [5] investigated various pointer acceleration curves used in common operating systems. They found that the default curve in OSX improved performance for small targets but reduced performance for large targets when compared to the curves used in Windows and Xorg. Similarly, Nancel et al. [7] studied

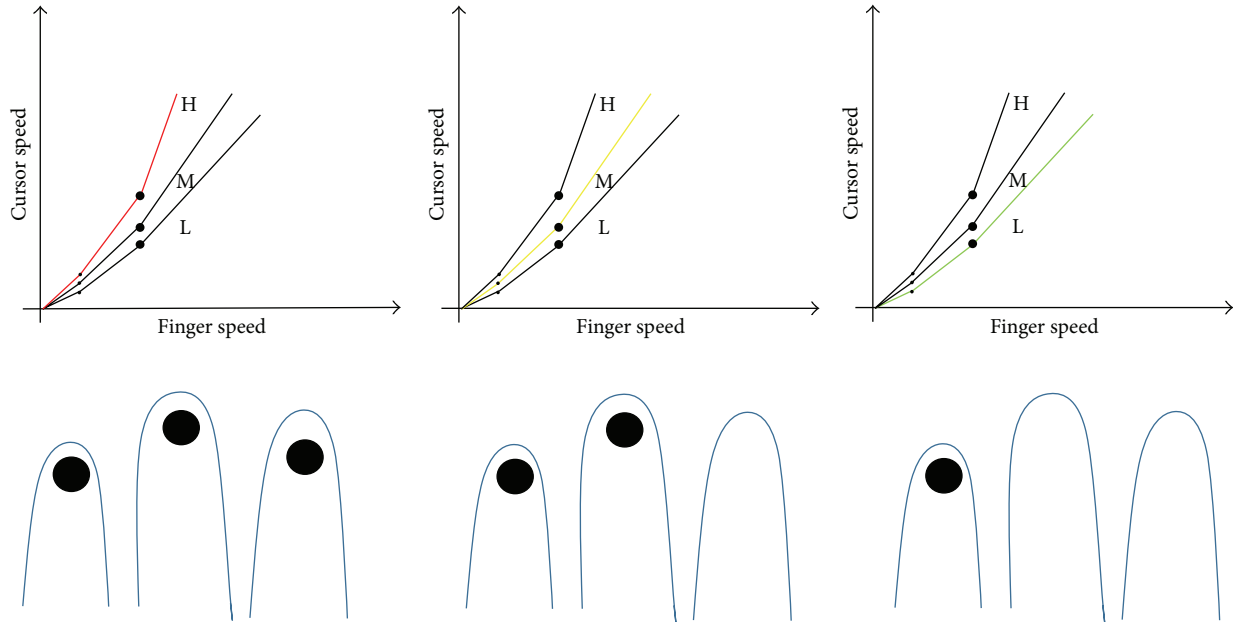


FIGURE 1: Conceptual image depicting the mechanism of switching pointer acceleration curves by varying the number of fingers in contact with the touchpad.

the utility of an optimized pointer acceleration curve for large high-resolution displays and concluded that such an acceleration curve performed poorly with targets of size less than 7 mm. Large displays require a large range of CD gains, thus requiring a sharply exaggerated acceleration curve that is overly sensitive to minor speed changes and difficult for users to control [4].

This shortcoming introduces the potential need for dynamic variation of the acceleration curve itself. Large high-resolution screens create a multiscale targeting environment where efficient continuous pointer positioning requires a diverse range of CD gains and acceleration curves for all scales of interaction that are possible on the display. Furthermore, since targeting conditions can rapidly vary in levels of scale, acceleration curves should be dynamically changeable in real time.

Additional interaction techniques can give users *explicit* granularity controls to change the acceleration curve. For example, Windows OS allows users to apply a multiplier (“pointer speed” in the control panel) to the acceleration curve [8], but this is a global setting not intended for dynamic use during tasks. Other examples, such as Nancel [4] and Dasiyici [6], attempt to accommodate the multiscale environment by allowing users to manually alternate between predefined curves by a mode switching mechanism. Their studies indicate that the interaction with the switching mechanism created a cognitive and a physical barrier that negatively affected performance.

1.1. Research Challenges. These issues introduce research questions about the inherent tradeoffs between the use of single or multiple pointer acceleration curves on large displays and how to effectively combine the advantages of an

efficient implicit acceleration curve with the power of explicit curve switching. Multiple curves can potentially enable a broader range of multiscale targeting conditions on large displays. However, these benefits can only be realized with an efficient mechanism for curve switching.

The main design challenge with an interaction technique that features dynamically switchable pointer granularity settings is to ameliorate the cognitive and physical obstacles introduced by the mechanism of switching as well as the mental dissonance associated with the process of switching between different modes of interactions. This raises a question of what interaction mechanism to use that would minimize the cognitive and physical efforts while allowing on-demand, dynamic, and instantaneous switching of pointer granularity.

1.2. Contribution. To address the problem of continuous pointing on large high-resolution displays (LHRD), providing both fast and accurate pointing with minimal cognitive barriers, we investigate the use of multiple (specifically, three) pointer acceleration curves to combine the benefits of implicit and explicit pointer acceleration. We propose design goals for the approach, suggest a specific technique implementation, and empirically study the tradeoffs of this multicurve approach against the single-curve approach. The design goals are as follows.

- (i) Exploit the implicit pointer acceleration curve for its proven usability.
- (ii) Augment with explicit switching among multiple (several) pointer acceleration curves to enable scalability.

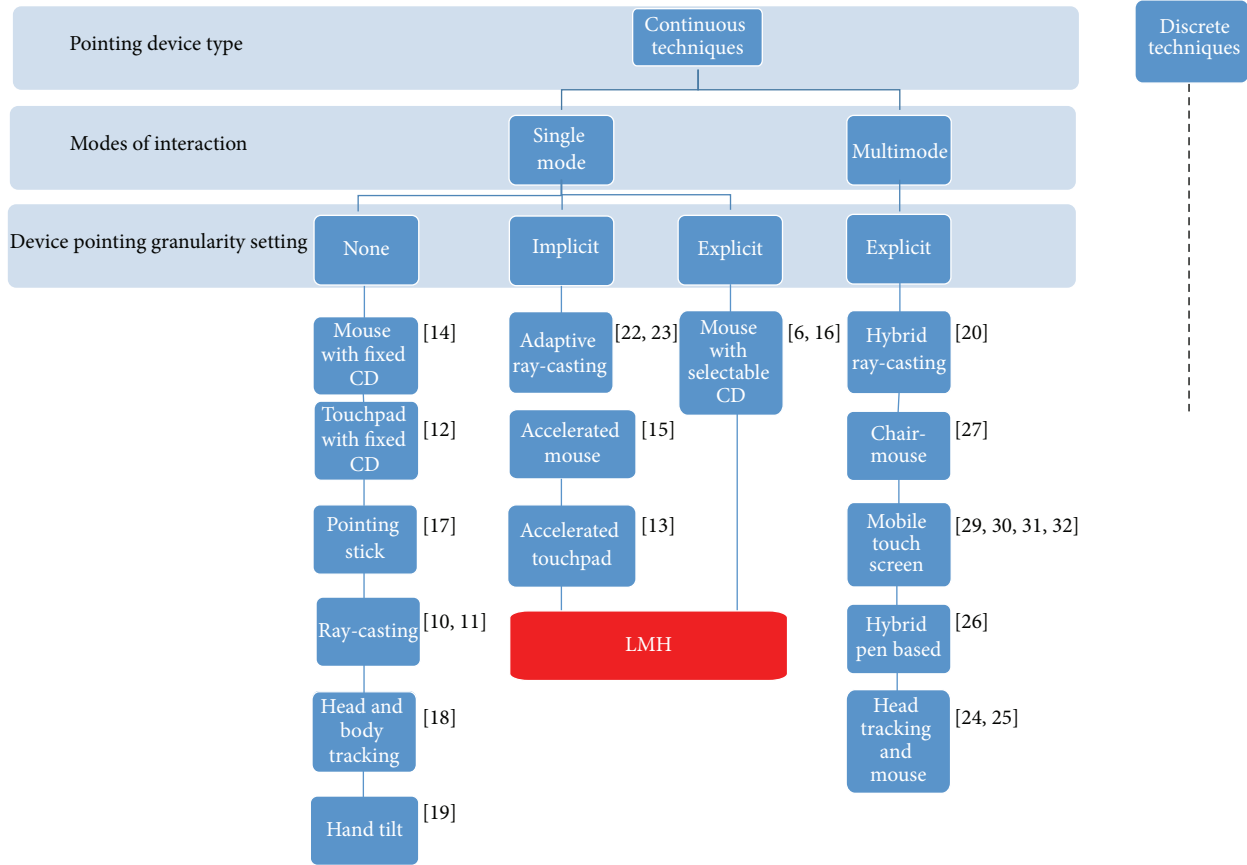


FIGURE 2: Categories of pointer interaction techniques for large displays, with prominent examples in each category. Multifinger LMH defines a new category of techniques that combine both explicit and implicit pointer granularity controls.

- (iii) Apply a single mode of interaction for both of the above, so as to enable the most rapid and fluid switching mechanism possible.

As a sample implementation of this approach, we design an interaction mechanism that seeks to minimize the effort required for explicit switching among curves. To minimize cognitive and physical effort, implicit pointer manipulation and explicit multicurve selection are incorporated cleanly into a single mode of interaction. Specifically, the design leverages the multitouch capabilities of modern touchpad devices to enable selection of different pointer acceleration curves by simply altering the number of fingers in contact with the capacitive surface of the touchpad while manipulating the pointer (as depicted in Figure 1).

This technique, called multifinger pointing, can allow users to dynamically select up to five different pointer acceleration curves by using one to five fingers to move the pointer. The number of fingers adjusts a multiplier applied to a standard acceleration curve. Thus, users can easily shift the range of CD gains that the acceleration curve offers. With up to five curves, this technique scales well beyond simple single- or dual-curve techniques, supporting very large displays and diverse targeting tasks. However, it may be impractical to scale beyond three fingers and curves.

We conduct a study to explore how users leverage this ability to dynamically switch acceleration curves as they manipulate the pointer and to compare the performance tradeoffs of multicurves against the standard single-curve technique. In the study, we test a 3-finger implementation so as to examine significant scalability while also maintaining a tractable study design. One finger has the lowest acceleration curve multiplier (L) which offers optimal CD gain range for small regions and targets; two fingers have a greater multiplier (M) that offers CD gain range for medium distances and targets; finally, three fingers produce the highest (H) multiplier and CD gain range that is aimed at coarse long distance movements. For convenience we refer to this specific implementation as LMH.

2. Current Techniques

The challenges of large display interactions are acknowledged and addressed through a diverse range of input devices and design philosophies [9] (Figure 2).

2.1. Single Mode Techniques. Techniques that allow continuous pointing without switching from one type of interaction to another can be dubbed as single mode techniques. Single mode techniques maintain the same mental perception of

the interaction technique even when the CD gain is changed. These techniques include ray-casting [10, 11], touchpads [12, 13], mice [6, 14–16], pointing sticks [17], head and body tracking [18] as well as hand tilting [19].

2.1.1. No Device Granularity Control. Techniques with no granularity control provide only a fixed CD gain. Absolute pointing techniques such as ray-casting [10, 11] essentially provide a CD gain of 1. Ray-casting allows users to interact with the display from any distance with simple wrist movements in the way that laser pointers are used. However, since users can change their distance to the display to vary their pointing accuracy, this might be considered a variable CD gain with explicit control (distance from display). The advantages of ray-casting include a natural interaction metaphor, low cognitive load, and freedom of movement in the display space. However, this interaction technique fails to provide high precision pointing [20, 21], especially with displays that feature high pixel density. The fatigue associated with manipulating the device results in involuntary hand tremors, which in turn reduces pointing accuracy.

2.1.2. Implicit Device Granularity Control. Some single mode techniques change the CD gain depending on the physical input from the user. The objective of implicit granularity control of the input device is to determine the targeting condition through the physical input from the user and then provide suitable CD gain for the inferred targeting condition.

For instance, adaptive ray-casting techniques [22, 23] allow variable CD gain to better match targeting conditions. With such techniques the CD gain changes depending on the hand movement characteristics and, as a result, provides faster and more precise continuous pointing experience. While the single mental model of ray-casting is preserved, the pointing performance improves.

The traditional mouse and touchpad provide very comfortable pointing experience with standard sized displays, with high accuracy and speed, and minimal physical and cognitive load. Modern operating systems provide dynamically variable CD gain, which enables accuracy at low speeds and accelerated pointer movement at higher speeds [8]. As a result, the mouse and touchpad have become ubiquitous pointing devices for normal displays. However, implicit granularity control, while being very useful on standard screens, fails to scale up to the diverse pointing conditions on LHRDs [5].

2.1.3. Explicit Device Granularity Control. Single mode interaction techniques with explicit granularity control allow users to maintain the same mental model for the interaction while providing the ability to explicitly change pointing characteristics of the input device. Extending the single interaction mode to accommodate multiscale targeting, by adding explicit controls, has the potential to not only provide high speed and accuracy but also eliminate the need to readjust to different pointing modes when interacting with LHRDs [2].

Single mode interaction techniques with explicit granularity control have had surprisingly little attention in the research community. In 1990, Jellinek and Card [16] introduced a modified mouse (manual mouse) that allowed users to change CD gain on the fly by pressing hardware buttons. In their study [16], they compared manually accelerated and constant gain mice. Unlike our experiment, Jellinek’s manual mouse enabled users to choose among fixed CD gains instead of acceleration curve multipliers. Their metaphor for the manual mouse was a manual gearbox with which drivers explicitly select their own gears, in contrast to an automatic gearbox that implicitly selects gears based on the speed of the vehicle. Although manual selection of CD gains provides support for a multiscale environment within a single interaction technique, Jellinek’s implementation suffered from a crude method of switching between different CD gain values.

In terms of Jellinek’s gearbox metaphor, our LMH technique would be analogous to a drivetrain with continuously variable transmission (CVT) and a differential with dynamically selectable ratios. It combines speed-based CD gain change (pointer acceleration) with the ability to instantaneously amplify the acceleration multipliers by varying the number of fingers touching the surface of the touchpad. Metaphorically speaking, the CVT is analogous to the pointer acceleration curve, while the differential ratio is analogous to the acceleration multiplier determined by the number of fingers. Thus, LMH users potentially benefit from both implicit (acceleration curve) and explicit (switching between different curves) granularity control as in Figure 3.

Furthermore, our LMH technique improves on Jellinek’s mechanism for explicit switching by using modern multi-touch touchpad devices to incorporate explicit controls into a single mode device. While most interaction techniques for LHRDs focus on multimode interactions, the issues associated with the process of switching between modes and the two modes interfering with each other are likely to remain. LMH addresses this issue by cleanly integrating the multiscale nature of the interaction technique into a single mode of interaction. Thus, we hope to minimize cognitive barriers that would normally occur during the transitions between different modes of interaction.

2.2. Multimode Techniques. With multimode techniques for continuous pointing, users switch between modes of interaction to handle different targeting conditions. Different modalities are separately exploited to independently support either fast or accurate pointing.

For example, Eye-gaze and head tracking techniques allow coarse positioning, which can be combined with traditional mouse or touchpad for precise target acquisition [24, 25]. Ray-casting is improved when precise positioning is added. Specifically, Vogel and Balakrishnan [20] introduce an interaction technique that allows switching between absolute pointing for coarse positioning and relative pointing for precision. Conversely, Forlines et al. [26] enable switching between absolute pointing with a pen for precision and relative pointing with high CD gain for coarse positioning. Endert et al. [27] captured natural body movements in a

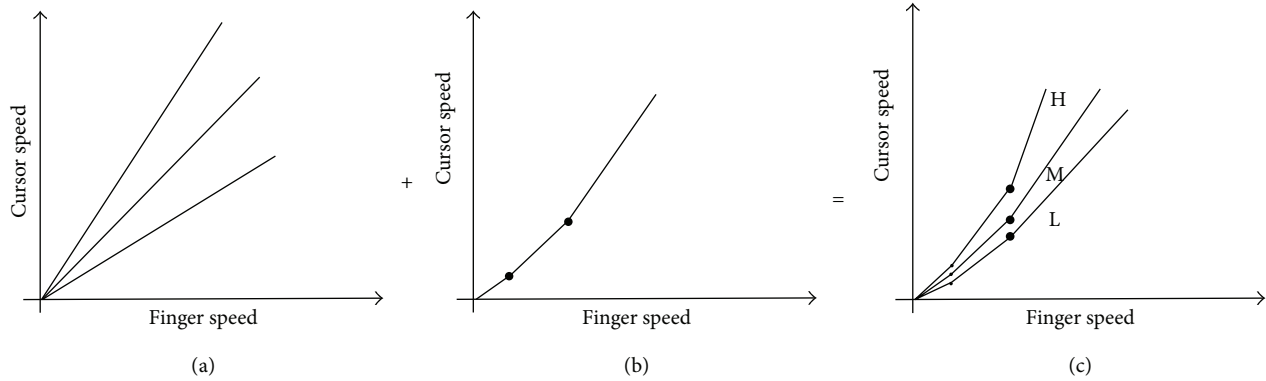


FIGURE 3: The LMH interaction technique (c) combines the advantages of explicit (a) granularity switching with the benefits of implicit accelerated (b) mice and touchpads.

rotating chair to enable coarse positioning and combined it with precise positioning using a traditional mouse. Zhai et al. [28] combined eye tracking with precise pointer stick positioning.

Malik et al. [29] introduced a multifinger gestural input with multitouch touchpad for interaction with large displays. The input technique enables coarse and fine positioning by mapping the left half of the touchpad to the entire screen and right half to a small rectangular area previously positioned using the left half (analogous to overview + detail). Effectively this approach allows direct manual control of the scale of interaction. Users switch between the two parts of the touch surface to alternate between scales.

Casiez et al. introduced RubberEdge [30] to reduce clutching by combining position and rate controls. The technique combines a touchpad with an elastic rate-control ring around the edges of the touchpad. The elastic ring translates physical pressure into directional pointer movement, with pointer speed proportional to the pressure applied. This approach provides the ability to manually select between relative and rate-based positioning inputs. Comparisons with a standard pointer accelerated touchpad showed effectiveness of the technique.

ARC-Pad [31] is another technique that is aimed at improving touchpad performance with large displays. It uses a small resistive touchscreen that combines absolute and relative positioning. Users tap the touchscreen to perform coarse-grained absolute positioning; then for near target refinements they drag their finger for relative positioning. The successor of this technique [32] uses a capacitive screen to allow users to switch from absolute to relative mode by switching to two fingers. Similar to RubberEdge [30], this technique showed improvements over regular touchpads. Our study is similar in nature and goals to these studies.

Nancel conducted a study [4] that compared pointing with dual-mode techniques against a single mode technique on a LHRD. Three dual-pointing techniques were used: a variation of Vogel et al.'s free-hand pointing where users switch modes by holding a button, a laser pointer and gyro mouse with a button that switches modes in a likewise fashion, and a laser pointer and touchpad where modes

switch whenever users touch the surface of the touchpad. For the single mode pointing technique, they used a gyroscopic mouse with pointer acceleration. The results of the study showed that the dual-mode techniques outperformed the gyroscopic mouse in both consistency and speed. However, with the dual-mode techniques some users tended to stay in the coarse pointing mode, which resulted in poor performance. According to the subjective feedback, some of the participants found it cognitively taxing to perform the explicit mode switching with the three techniques. The process of switching modes with the three techniques was not instantaneous; the duration of the time required to switch modes exhibited a correlation with the distance between targets and with their sizes. The authors claim that some users tended to stay in one mode with the intention of performing faster pointing.

While multimode techniques do improve pointing performance on LHRDs, the process of transitioning between modes imposes a cognitive load on users. Additionally, differences between modes sometimes discourage [32] users from switching, and this has a negative effect on the performance. To overcome this problem, our LMH technique attempts to incorporate the ability to explicitly and instantaneously switch continuous pointer acceleration levels into a single interaction metaphor.

3. Design

Our main design goal is to enable users to dynamically alter pointer acceleration curve properties in a fluid way, minimizing the physical and cognitive barriers associated with manual switching. Our aim is to exploit the usability of implicit pointer acceleration curves but scale up the usable range of CD gains by also enabling explicit modification of the acceleration curve. The interaction mechanisms for these should be cleanly integrated, and so we explore the use of a single mode of interaction. We also seek to explore if this can be designed in a way that easily fits into existing technologies and usage.

Our proposed technique design, called multifinger pointing, is to dynamically alter acceleration curve properties by

simply varying the number of fingers contacting the surface of the touchpad while manipulating the pointer. Specifically, the number of fingers can be used to control a multiplier applied to the pointer acceleration curve during pointer manipulation.

The design rationale has several points. Various multi-finger gestures have gained popularity in modern computing systems, using multitouch touchpads and touchscreens. These gestures are typically used for tasks such as scrolling, zooming, application switching, dragging, and selecting [33]. Low cognitive load of the multifinger interactions [34] is the inspiration for the design of multifinger pointing. By associating higher speeds with more fingers on the touchpad, multifinger pointing offers users a way to externalize and associate particular targeting conditions with the number of fingers they have in contact with the surface. The natural real-world metaphor is that more fingers apply more force for coarse actions, while a single finger applies a light touch for fine adjustment. The interaction involves minimal additional effort since the fingers are already present and moving over the touchpad anyway, due to the use of wrist movement with touchpads. By overriding two and three finger swiping gestures for pointer manipulation, we exploit the low cognitive load of multifinger interaction to allow users to adapt pointer speeds to the diverse targeting conditions that arise on LHRDs. Of course, this interaction could conflict with existing use of multifinger gestures for other actions, which must be considered in the broader HCI design of LHRD systems. As a potential solution, since other actions such as scrolling involve a cognitive task switch anyway, pressing a function key to switch touchpad modes might be reasonable for those actions.

Other potential designs that achieve these design goals are possible. For example, a pressure sensitive touchpad could vary the acceleration curve multiplier with finger pressure, called pressure pointing. This design has very similar characteristics to multifinger pointing. Pressure-pointing could conflict with the existing use of pressure sensitive clicking.

3.1. Implementation. For the purposes of the experiment, we implement a specific version of multifinger pointing, called LMH. LMH allows users to choose any one of three pointer acceleration curve multipliers at any given time, by applying one, two, or three fingers to the surface of the touchpad. Users can add and release fingers during the course of finger movement to dynamically switch curves. When multiple fingers are applied, LMH uses the average speed of all contacting fingers. This captures the overall hand movement better than simply capturing the speed of any one finger.

We established three different pointer acceleration multipliers: L for low speed, M for medium speed, and H for high speed. These three multipliers are scaling factors applied to a standard pointer acceleration curve. In the study, we used $L = 0.7$, $M = 3.4$, and $H = 7.2$. These three values were identified through a focus group and informal usability experiments. Acceleration multipliers are applied to an acceleration curve that was optimized for the touchpad and the LHRD system in the experiment. For the pointer acceleration curve we used

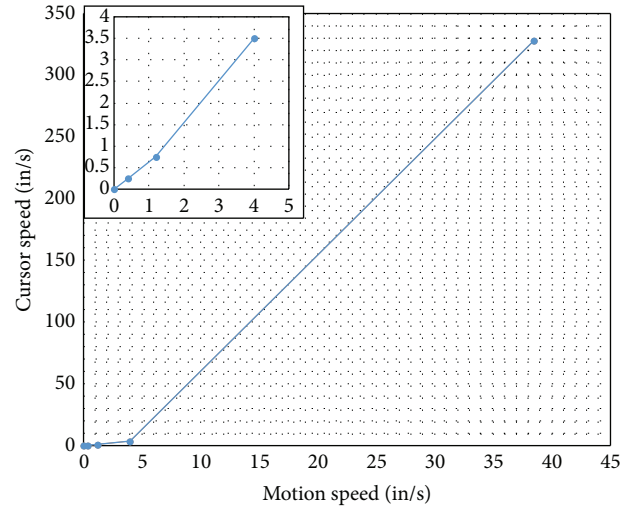


FIGURE 4: The pointer acceleration curve used in the experiment. The figure depicts the curve with a multiplier of 1.0. The curve represents the relationship between input speed and output speed. A CD gain curve can be derived by dividing each point on the curve by its input speed.

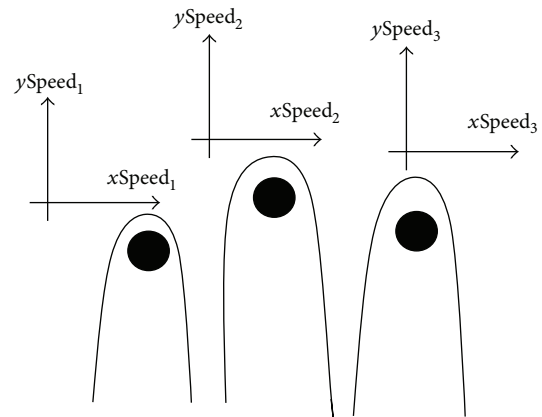


FIGURE 5: The input velocity vector is the average of the velocity vectors of all fingers that are in contact with the surface.

a modification of the Windows XP curve [8] as shown in Figure 4.

LMH is implemented using the TUIO [35] open framework which is commonly used for developing multitouch user interfaces. We used the TUIO API to implement the client software that is responsible for pointer manipulations on the LHRD. The client on the LHRD system receives input from a MacBook Pro over a WiFi network. To capture multitouch events on the laptop's touchpad, we employed the Tongseung [36] TUIO event wrapper which sends the TUIO events to the TUIO client.

The acceleration curve is applied to the average of the velocity vectors of all the fingers contacting the touchpad as follows. Whenever fingers move on the touchpad, TUIO reports each finger independently with its own velocity vector (Figure 5). LMH computes the overall control velocity vector

as the average of the velocity vectors of all fingers at the current timepoint (see (1)). Then, LMH computes the CD gain on the magnitude of the control velocity vector using the pointer acceleration curve (Figure 4) and applies the multiplier (L, M, or H) depending on the number of fingers currently contacting the touchpad as reported by TUIO (see (3)). The CD gain is then applied to the control velocity vector to compute the display pointer velocity vector, which is used to displace the pointer on the screen. Consider

$$\overrightarrow{\text{control Velocity}} = \frac{1}{\text{numFingers}} \sum_{i=1}^{\text{numFingers}} \overrightarrow{\text{finger Velocity}_i}, \quad (1)$$

$$\text{multiplier}(\text{numFingers}) = \begin{cases} 0.7, & n = 1, \\ 3.4, & n = 2, \\ 7.2, & n = 3, \end{cases} \quad (2)$$

CD gain = multiplier (numFingers)

$$\times \frac{\text{pointer AccelCurve} \left(\left| \overrightarrow{\text{control Velocity}} \right| \right)}{\left| \overrightarrow{\text{control Velocity}} \right|}, \quad (3)$$

$$\overrightarrow{\text{pointer Velocity}} = \text{CD gain} \times \overrightarrow{\text{control Velocity}}. \quad (4)$$

4. Experiment

Our main goal in this study is to determine how users leverage the combination of implicit and explicit pointing granularity control to interact with multiscale tasks on a LHRD. We specifically observe how users utilize the ability to dynamically alter acceleration curve properties during task performance by simply varying the number of contacting fingers touching the surface of a touchpad. Does the addition of multiple acceleration curves into a single mode of interaction enable users to improve performance over a single acceleration curve? Does this access mechanism overcome physical and cognitive barriers introduced by explicit switching? What implicit versus explicit switching behaviors do users exhibit during pointing tasks of varying scale characteristics? To study and understand these issues, we compared multifinger LMH with the standard single-finger technique at each of the three fixed curve multipliers (L, M, and H).

4.1. Techniques. In this study we compared four techniques: multifinger LMH that uses three different acceleration curve multipliers (L = 0.7, M = 3.4, and H = 7.2) and three single-finger techniques with the same L, M, and H multipliers, respectively. Each of these multipliers represents different settings of the Windows XP “pointer speed” slider in the control panel, which scales the default pointer acceleration curve. Our hypothesis is that any of the individual settings will be insufficient to provide good task performance across the full range of task scales on a LHRD, but by efficiently enabling

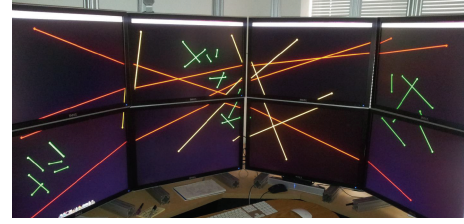


FIGURE 6: The LHRD used in the study. Red, yellow, and green lines demonstrate the concept of the LMH technique in terms of typical distances covered with each multiplier. H is used for large distances (red lines), M is used for medium distances (yellow lines), and L is for short distances (green lines).

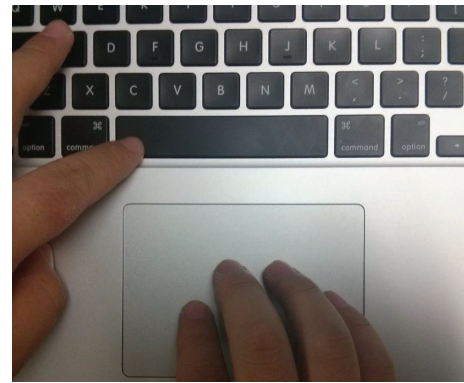


FIGURE 7: Users performed target acquisition tasks using the touchpad on a MacBook Pro laptop. Users performed selections (clicks) using the space bar of the laptop with the other hand so as to avoid interference with the pointer movement task.

dynamic switching between all three settings LMH will provide consistently good performance. Thus, we hypothesize that, overall, LMH will offer better performance than L, M, or H individually. We chose to apply multifinger pointing with 3 curves so as to demonstrate scalability to more than 2 curves and also to limit the number of experimental conditions to 4.

4.2. Hardware. For this experiment we used a large display that has a total resolution of 10240 × 3200 pixels, constructed using eight 30-inch monitors with 100 DPI in a 2 × 4 grid (Figure 6). For interaction, we used the multitouch touchpad of a 2010 MacBook Pro 13. The touchpad was used to control the pointer on the large display. One of the reasons we opted for the MacBook Pro’s touchpad is users’ familiarity with its ergonomics. The clicking was performed (Figure 7) with the spacebar button of the laptop instead of tapping. By doing so we kept pointer tracking performance isolated from potential interference of tapping or clicking the touchpad itself. This is another reason for choosing a laptop’s touchpad over a dedicated touchpad such as Apple Magic Trackpad.

4.3. Tasks and Procedure. In a within-subjects design, each participant performed all target tasks with all four interaction techniques. Eight undergraduate students participated in the experiment (three male and five female) between 19 and 23

years old. Participants were first given time to familiarize themselves with the hardware and interaction and given practice tasks at the beginning of each technique condition block. Then, participants were asked to perform the task trials as quickly and accurately as possible. The average duration of the experiment was 75 minutes per participant.

Two-dimensional reciprocal target acquisition tasks (Figure 8) with round targets were used, similar to Casiez et al. [30]. We used three target sizes and three distances, including small 3 mm targets at long distances over 2 m to cover the multiscale nature of LHRD tasks. Each participant performed a total of 216 trials (4 Techniques \times 3 Sizes \times 3 Distances \times 3 Blocks \times 2 Repetitions), with a total of 1728 trials across all 8 participants. Cases with missed clicks were marked as errors, and the number of such errors was recorded. The technique condition order of L, M, H, and LMH was counterbalanced in a Latin-square design across the 8 users. The presentation order of the target sizes and distances (see Table 1) was randomized.

We measured target selection time and number of clutches, as well as distance travelled by one, two, and three fingers. A clutch is defined as an instance of lifting all of the fingers from the touchpad. We recorded the number of clutching instances during each target acquisition task. We also recorded the distance travelled by the pointer and the paths taken by the pointer. We recorded the number of fingers used at all times, which allows us to measure individual contributions of each finger combination. The data was captured by the reciprocal target acquisition task software that we developed, by monitoring the TUIO finger events and pointer events within the software.

5. Results

5.1. Selection Time. Selection time results are shown in Figure 9. Analysis of variance with repeated measures demonstrated significant main effects on selection time for the technique ($F_{3,21} = 18.72, P < 0.0001$), distance between targets ($F_{2,14} = 422.43, P < 0.0001$), and target width ($F_{2,14} = 190.08, P < 0.0001$). There was a significant interaction effect of technique \times distance on selection time ($F_{6,42} = 57.17, P < 0.0001$). We detected significant overall differences through pairwise comparisons, with LMH faster than M ($P < 0.001$), LMH faster than H ($P < 0.0001$), and M faster than H ($P < 0.0001$), whereas L had high variance depending on the 3 distances, contributing to the interaction effect.

We now look at each of the three target distances in turn. For short distance targets, significant pairwise comparison revealed that LMH and L were faster than M and H, but L was 13% faster than LMH ($P < 0.017$). This is unexpected since LMH with one finger is interactively identical to L. For short distances, LMH users should only need to use the one-finger L curve. However, some users unnecessarily used more than one finger for some of these task trials, which had the effect of slowing their performance. For medium distance targets, pairwise comparison of LMH, L, and M revealed no significant differences. However, H was slower than the rest (all three $P < 0.0001$). For long distance targets, LMH was

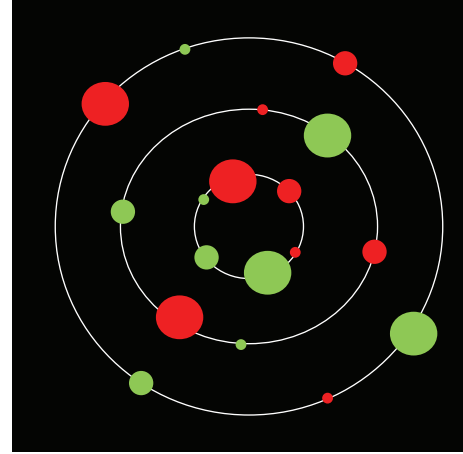


FIGURE 8: (This is not drawn to scale.) Reciprocal target acquisition task: users start by clicking the green target that appears at a random size and location on any of the rings. When the green target is clicked, the red target appears reciprocally to the green target. The measurements begin when the green target is clicked and end when the red target is clicked.

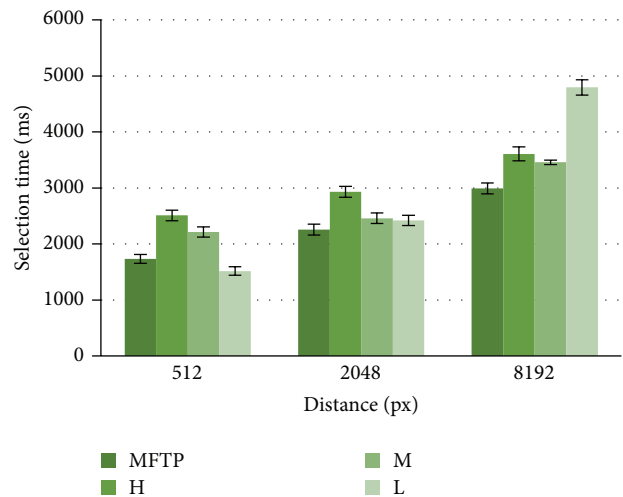


FIGURE 9: Mean selection time for each technique and target distance. Overall LMH is the most balanced technique across the distances (error bars: 95% confidence interval).

faster than all the others (all three $P < 0.0001$), and L was slower than all the others (all three $P < 0.0001$).

In general, these results validate the basic hypothesis that individually L, M, and H excel at different distance scales. In particular, L excelled at short distances but performed very poorly at long distances. M performed well at the longer distances. H appeared to be overkill, but given the trend it would likely do well at longer distances or larger targets than we tested.

Overall, LMH was well balanced, nearly satisfying the goal of being as fast as all single-curve techniques at all distances. Its only failing was to fall slightly behind L at the shortest distances due to the unnecessary use of multiple fingers in those cases. This perhaps indicates that there is

TABLE 1: The target sizes and distances used in the study (in pixels and millimeters).

Target size		Distance	
px	mm	px	mm
12	3.05	512	130.05
24	6.1	2048	520.19
48	12.19	8192	2080.77

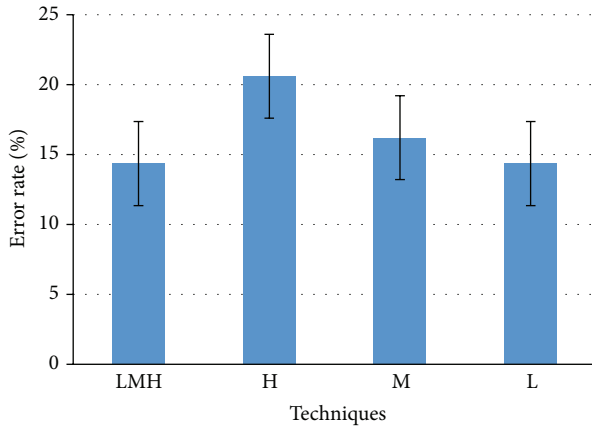


FIGURE 10: Mean error rate for each technique (error bars: 95% confidence).

still some cognitive struggle in explicitly selecting among multiple curves. It seems natural that, with more training and experience, users could overcome this problem by simply recognizing that a single finger is sufficient for short distance operations. Considering L's extremely poor performance at long distances, LMH is the best overall choice. M is the next most balanced but is slower than LMH for short and long distances.

5.2. Error Rates. The error rate was high (Figure 10), which is not surprising given the difficulty of a multiscale task of selecting small targets over long distances. Technique had no significant main effect on the error rate ($F_{3,21} = 1.97$, $P < 0.117$). Several participants noted that positioning the pointer on the smallest target (3 mm) was difficult to precisely see on the screen. The trend indicates that H suffered higher error rates than L and LMH (both $P < 0.037$). The advantage of L and LMH over H is obvious since both of them are optimized for precision. It also makes sense that L and LMH are essentially identical in error rates, since both use the same L curve for precision pointing. This indicates that LMH users did make use of the 1-finger option for precision. Taken together with the selection time results, LMH enables fast performance while also supporting precision.

5.3. Clutching Rates. Clutching is when users lift all fingers from the touchpad during a pointing task and usually occurs when the user's fingers reach the edge of the touchpad and must be lifted and repositioned on the touchpad to make room for further pointer movement. Clutching negatively

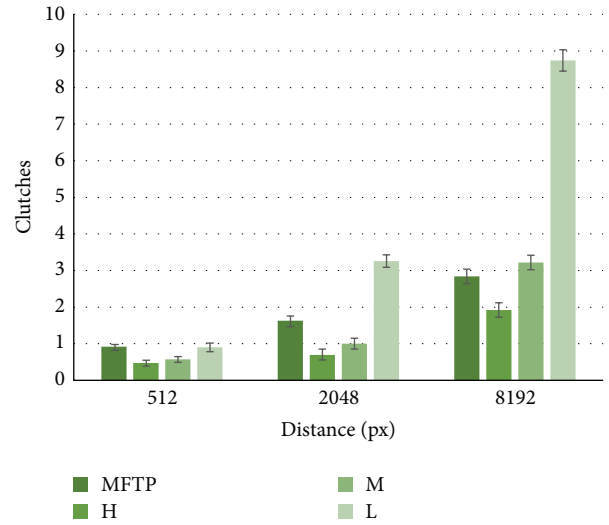


FIGURE 11: Clutching rates for each technique and each target distance (error bars: 95% confidence range).

affects user performance in pointing tasks [30]. Thus, we compare the number of occurrences of clutching for different techniques (Figure 11). The average number of clutches per task for the experiment was 2.2. We found significant main effects on clutching by technique ($F_{3,21} = 69.97$, $P < 0.0001$) and distance between targets ($F_{2,14} = 125.34$, $P < 0.0001$) and an interaction effect of distance \times technique ($F_{6,42} = 94.01$, $P < 0.0001$).

For long distances, H had the fewest number of clutches (mean 1.9), followed by LMH (2.9), M (3.2), and L (8.7). The higher clutch rate of L for long distances is predictable given the low acceleration curve multiplier. H had 33% fewer clutches than LMH, despite the fact that LMH's "top gear" uses the identical multiplier for the acceleration curve as H. Theoretically, LMH should be able to support as few clutches as H, since LMH users can switch multipliers without lifting all fingers. Yet, we observed some participants clutching when they switched to using fewer fingers for more precision after getting near the target. This form of clutching with LMH seems to occur at all distances. Contrasting the clutching results with the target selection time results shows that LMH performs better than H despite the higher clutch rate, indicating that taking the time to switch curves is worthwhile.

5.4. Target Acquisition Patterns with LMH. The most efficient use of LMH requires switching curves depending on the targeting conditions. For quick long distance targeting, high speeds should be used to reach the general area of the target; once the pointer is close to the target, lower speeds should be used for accurate target acquisition. Here we analyze whether users employ LMH in this ideal fashion. To analyze the usage patterns of LMH, we recorded the path of the pointer as it is moved between targets and the number of fingers touching the surface of the touchpad at each timepoint along the path. With this data we can visualize the patterns of LMH usage (Figure 12) for various target distances.

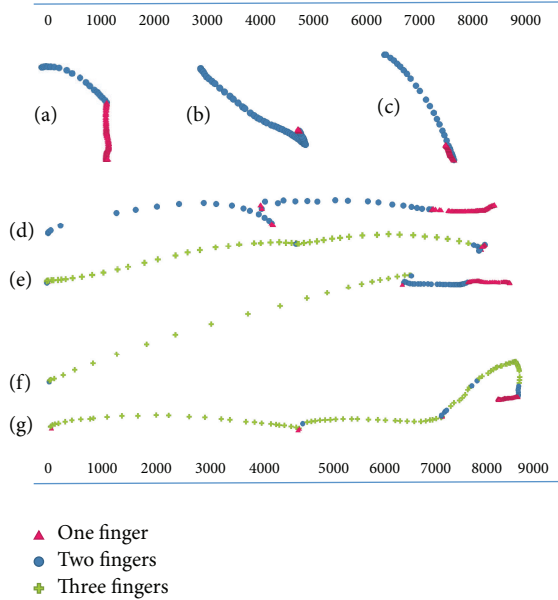


FIGURE 12: Representative examples of pointer movement patterns for medium ((a)–(c)) and long distance ((d)–(g)) targets. The gaps between points captured at a constant sample rate are proportional to the speed of the pointer. Scales show pixel distances.

Short Distance. For the short distance targets, participants mostly used one-finger mode (low speed curve). However, in 18% of cases, 5 out of 8 participants used more than one finger, which explains the difference in performance between L and LMH for short distances. Postexperiment discussion revealed that only one participant knowingly used two fingers for the short distance targets. The participant incorrectly believed that with two fingers it would be faster to acquire the targets.

Medium Distance. For medium distance targets (Figures 12(a), 12(b) and 12(c)), all LMH users employed a combination of one and two fingers. They usually started with two fingers (medium speed curve) and, once near the target, switched to one finger (low speed curve). Some users switched to one finger prematurely (Figure 12(a)). Some were accurate with two fingers (Figure 12(b)). We observed some pointer overshooting with two fingers, followed by one-finger fix (Figure 12(c)).

Long Distance. For long distance targets (Figures 12(d), 12(e), 12(f) and 12(g)), most LMH users exhibited consistent use of three fingers. On average, three fingers were used to cover 84% of the pointer’s total trip to the target. One and two fingers contributed only 9% and 7%, respectively. However, only one participant deliberately used only one or two fingers (Figure 12(d)), ignoring the three-finger mode. Pointer overshooting was observed for some of the participants, especially during the practice trials as users accustomed themselves to LMH. We believe that overshooting is primarily attributed to the fact that users need some time to learn how to smoothly switch out of the three-finger mode as the pointer approaches the target. For the entire experiment, three general patterns

emerged: (1) “spot on” pattern (Figure 12(e)) with straight trajectory, minimal use of slow speeds, and minimal overshooting, (2) “gradual descent” pattern (Figure 12(f)) with straight trajectory, prolonged use of slower speeds on approaching the target, and no overshooting, and (3) “zig-zag” pattern (Figure 12(g)) with a bent trajectory, overshooting at high speed, and late use of slower speeds to hone in on the target.

From these results it is clear that most users did make good use of the ability to explicitly choose among multiple acceleration curves. They did appropriately adjust their usage patterns according to the varying multiscale targeting conditions of each task, using primarily three fingers (H) for the long distance tasks, two fingers (M) for medium distance, and one finger (L) for short distances. They did dynamically switch curves during the course of individual targeting tasks. They primarily used a “high-to-low” pattern, switching to fewer fingers during the course of a targeting task to hone in on the target, similar to the theoretical ideal pattern. They almost always finished tasks with a single finger (L) for final refinements. There was variance in the margin within which they switched down to a lower multiplier, including some undershooting and some overshooting. The usage patterns indicate that they tended to clutch when switching curves, and vice versa, suggesting that more training time might enable them to use the technique more smoothly with less clutching.

5.5. User Feedback. In a postexperiment questionnaire, all participants preferred LMH most out of the four techniques. In verbal feedback and discussion, they cited LMH’s adaptive nature. They appreciated the ability to slow down near the targets by lifting extra fingers. Some said that “speed” switching required no noticeable effort and was performed “without thinking about it.” When asked to describe their target acquisition strategies, all of them mentioned switching to the “slow speed” when approaching a target. Several participants described LMH as “fun” and “engaging.” Seven out of the eight participants said that they would use LMH if they had a display as large as the one used during the experiment. Five participants said that they found LMH intuitive after the first dozen clicks. The other three also described it as intuitive, although during the first several minutes of using LMH they had the opposite opinion.

Among the single-finger techniques, participants preferred M over H, since it was easier for precise positioning due to the lower acceleration multiplier, although they described it as somewhat uncomfortable. For both H and M, all users verbally expressed frustration when using them, especially for H. Small target acquisition was challenging due to high CD gains, and participants said that precise control required significant careful effort. However, the worst technique according to participants was L, because of the large amount of clutching required for long distance target acquisition. With L, all participants said they had no difficulty acquiring targets, but the numerous clutching for long distance pointer movements caused them fatigue, although, for short distances, they preferred L due to its low CD gain which makes it easy to select small targets.

On the questionnaire, participants were also asked to rate the difficulty of each technique on a scale from 1 to 7, with higher number indicating greater difficulty. While the results of the questionnaire resonate with the results of the informal interviews, the numerical results are not statistically significant (Figure 13). It is interesting that participants rated L and H similarly and rated them both as more difficult than M and LMH. This indicates that users find it difficult to use techniques that either are inaccurate (H) or require too much clutching (L). They preferred the more balanced techniques.

6. Discussion and Future Work

Does the addition of multiple acceleration curves into a single mode of interaction enable users to improve performance over a single acceleration curve? This study confirms that users were able to use three pointer acceleration curves in a way that improved their overall performance over using only any one of the curves. That is, users exploited explicit curve switching (among three curves) to greatly expand the range of CD gains available to them in order to efficiently accomplish multiscale tasks on a LHRD. Users utilized the higher multipliers (H) for fast coarse positioning over long distances and the lowest multiplier (L) for fine positioning over small nearby targets. Thus, while the single-curve techniques exhibited the expected tradeoff at different distances, LMH broke the tradeoff. Importantly, the LMH technique enabled the fastest long distance selection times, while also matching the accuracy (low error rates) of the low speed curve (L). This combination of explicit and implicit control of pointer granularity yielded better overall results than strictly implicit control. The only exception was slightly slower performance than L alone for the shortest distance targets (discussed later).

Does the multifinger access mechanism overcome physical and cognitive barriers introduced by explicit switching? This evidence indicates that the LMH multifinger pointing technique did overcome much of the physical and cognitive barriers associated with explicit curve switching by preserving the same model of interaction across scales. Participants said they found it easy to use and natural to dynamically switch settings, and their usage patterns indicate that they did fluidly switch the number of fingers during the course of each task in helpful ways. The fact that users performed as well as they did with minimal training is evidence of the natural fit of the multifinger technique.

What implicit versus explicit switching behaviors do users exhibit during pointing tasks of varying scale characteristics? The target acquisition patterns show that users consistently utilized explicit curve switching in LMH in a way that is similar to the designed optimal usage pattern, beginning with two and three finger swiping for coarse pointing, followed by single finger precise refinement. Only one user did not consistently use all three curves.

However, we found two main problems in the results that lead to potential future work. These two problems indicate that there is still some vestige of physical and cognitive barriers to explicit switching in the LMH multifinger pointing

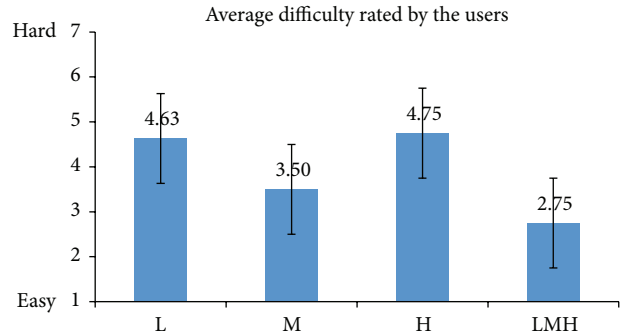


FIGURE 13: Averages of the user rated difficulty for each technique on a scale from 1 to 7, where higher numbers mean more difficulty.

technique. First, as a physical barrier, users tended to perform slightly more clutching than was necessary. It seems that some users occasionally clutched when explicitly switching curves, which may have slowed their overall performance. Second, as a cognitive barrier, there was some use of suboptimal strategies in the patterns of explicit switching. For example, some users attempted to unnecessarily use two fingers for short distance targets, which explains the slightly slower performance of LMH over L for those tasks. Also, one user avoided using three fingers even for long distance targets. There was some variance in the timing of the explicit switching from H to M to L to hone in on targets, causing some users to undershoot or overshoot. Yet, these problems do not appear to be fundamental and could possibly be overcome simply with more training and experience by the users, or refinements to the design.

To further reduce physical and cognitive costs of dynamically switching CD curves, refined design ideas for integrating these concepts into a single mode of interaction can be explored in future work, such as using pressure sensitive touchpads. The core concept of cleanly combining implicit and explicit granularity control into a single mode of interaction can be applied to other kinds of input devices, such as “Smarties” [37] that uses mobile touchscreen devices for wall display interactions. Alternatively, the number of fingers touching the surface of a mouse could act as a trigger to change pointer acceleration multipliers, similar to LMH. Furthermore, beyond varying the number of touching fingers, other input mechanisms could be employed, such as distance [38] from the screen. Finally, these integrated curve-switching techniques should be compared with dual- or multimode techniques that link the various CD curves to different modes of interaction (e.g., ChairMouse [27]) to more closely examine the effects of interaction modality.

Other useful future work would include examining how the multifinger pointing technique would scale up to even higher resolution displays and additional acceleration curves. It would also be interesting to conduct longer-term studies in more complex tasks (e.g., [39]) to go beyond any effects of novelty and reveal more prominent performance insights into how users utilize implicit and explicit pointer granularity controls.

7. Conclusion

This paper makes three contributions. First, we identify a new class of interaction techniques for LHRDs that combine implicit and explicit pointer acceleration controls within the same mode of interaction. Second, we propose a new interaction technique design within this new class, called multifinger pointing, that allows users to dynamically choose from up to five acceleration curves while manipulating the pointer by simply changing the number of fingers in contact with the multitouch touchpad. Our implementation, LMH, supports switching between three curve multipliers. Third, we find through an empirical study that the combination of implicit and explicit controls in the form of multifinger LMH improved overall performance when compared to just implicit control (L, M, or H), enabling both fast long distance movement and accurate small target refinement. Participants found it easy to use and preferred it. Participants appropriately used the ability to explicitly switch pointer acceleration curves, depending on targeting conditions, to efficiently perform multiscale tasks on a large high-resolution display. We hope this work spurs on new designs for single-mode interaction techniques that cleanly integrate multicurve pointer acceleration into existing interfaces for large high-resolution display systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] R. Ball and C. North, "Effects of tiled high-resolution display on basic visualization and navigation tasks," in *Proceedings of the Extended Abstracts on Human Factors in Computing (CHI EA '05)*, pp. 1196–1199, ACM, Portland, Ore, USA, April 2005.
- [2] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, and J. E. K. Smith, "Optimality in human motor performance: ideal control of rapid aimed movements," *Psychological Review*, vol. 95, no. 3, pp. 340–370, 1988.
- [3] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn, "The impact of control-display gain on user performance in pointing tasks," *Human-Computer Interaction*, vol. 23, no. 3, pp. 215–250, 2008.
- [4] M. Nancel, *Designing and combining mid-air interaction techniques in large display environments [Ph.D. thesis]*, Université Paris Sud-Paris XI, 2012.
- [5] G. Casiez and N. Roussel, "No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 603–614, October 2011.
- [6] M. C. Dasiyici, *Multi-Scale Cursor: Optimizing Mouse Interaction for Large Personal Workspaces*, Virginia Polytechnic Institute and State University, 2008.
- [7] M. Nancel, E. Pietriga, and M. Beaudouin-Lafon, "Precision pointing for ultra-high-resolution wall displays," Tech. Rep. RR-7624, INRIA, 2011.
- [8] Microsoft, Pointer-Ballistics for Windows XP, 2012, <http://www3.ntu.edu.sg/home2009/TANJ0181/fyp.html>.
- [9] B. Badillo, D. A. Bowman, W. McConnel, T. Ni, and M. G. da Silva, "Literature survey on interaction techniques for large displays," Tech. Rep. TR-06-21, Computer Science, Virginia Tech, 2006.
- [10] R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg, "A comparison of ray pointing techniques for very large displays," in *Proceedings of the Graphics Interface (GI '10)*, pp. 269–276, Ottawa, Canada, 2010.
- [11] C. Kirstein and H. Muller, "Interaction with a projection screen using a camera-tracked laser pointer," in *Proceedings of the Conference on MultiMedia Modeling (MMM '98)*, pp. 191–192, 1998.
- [12] R. G. Ball, R. S. Newton, and D. Whitfield, "Development of an off-display, high resolution, direct touch input device: the RSRE touchpad," *Displays*, vol. 1, no. 4, pp. 203–207, 1980.
- [13] G. J. Anderson, "Speed zone touchpad," Google Patents, 2002.
- [14] S. K. Card, W. K. English, and B. J. Burr, "Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT," *Ergonomics*, vol. 21, no. 8, pp. 601–613, 1978.
- [15] D. V. Keyson, "Dynamic cursor gain and tactual feedback in the capture of cursor movements," *Ergonomics*, vol. 40, no. 12, pp. 1287–1298, 1997.
- [16] H. D. Jellinek and S. K. Card, "Powermouse and user performance," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*, pp. 213–220, 1990.
- [17] D. C. Burson, *System with Joystick to Control Velocity Vector of a Display Cursor*, Google Patents, 1979.
- [18] K. Nickel and R. Stiefelhagen, "Pointing gesture recognition based on 3D-tracking of face, hands and head orientation," in *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*, pp. 140–146, Vancouver, Canada, November 2003.
- [19] M. Rahman, S. Gustafson, P. Irani, and S. Subramanian, "Tilt techniques: investigating the dexterity of wrist-based input," in *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems (CHI '09)*, pp. 1943–1952, April 2009.
- [20] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*, pp. 33–42, Honolulu, Hawaii, USA, October 2005.
- [21] B. A. Myers, R. Bhatnagar, J. Nichols et al., "Interacting at a distance: measuring the performance of laser pointers and other devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves*, pp. 33–40, 2002.
- [22] L. Gallo, M. Ciampi, and A. Minutolo, "Smoothed pointing: a user-friendly technique for precision enhanced remote pointing," in *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '10)*, pp. 712–717, February 2010.
- [23] W. A. König, J. Gerken, S. Dierdorf, and H. Reiterer, "Adaptive pointing—design and evaluation of a precision enhancing technique for absolute pointing devices," in *Human-Computer Interaction—INTERACT 2009*, pp. 658–671, Springer, 2009.
- [24] M. Ashdown, K. Oka, and Y. Sato, "Combining head tracking and mouse input for a gui on multiple monitors," in *Proceedings*

- of the Conference on Human Factors in Computing Systems, pp. 1188–1191, April 2005.
- [25] S. Zhai, *Human Performance in Six Degree of Freedom Input Control*, University of Toronto, Ontario, Canada, 1995.
 - [26] C. Forlines, D. Vogel, and R. Balakrishnan, “HybridPointing: fluid switching between absolute and relative pointing with a direct input device,” in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*, pp. 211–220, October 2006.
 - [27] A. Endert, P. Fiaux, H. Chung, M. Stewart, C. Andrews, and C. North, “ChairMouse: leveraging natural chair rotation for cursor navigation on large, high-resolution displays,” in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI '11)*, pp. 571–580, May 2011.
 - [28] S. Zhai, C. Morimoto, and S. Ihde, “Manual and gaze input cascaded (MAGIC) pointing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 246–253, 1999.
 - [29] S. Malik, A. Ranjan, and R. Balakrishnan, “Interacting with large displays from a distance with vision-tracked multi-finger gestural input,” in *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, pp. 43–52, October 2005.
 - [30] G. Casiez, D. Vogel, Q. Pan, and C. Chaillou, “RubberEdge: reducing clutching by combining position and rate control with elastic feedback,” in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*, pp. 129–138, October 2007.
 - [31] D. C. McCallum and P. Irani, “ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen,” in *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*, pp. 153–156, October 2009.
 - [32] M. Nancel, O. Chapuis, E. Pietriga, X.-D. Yang, P. Irani, and M. Beaudouin-Lafon, “High-precision pointing on large wall displays using small handheld devices,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 831–840, May 2013.
 - [33] Apple, Multi-touch gestures, 2012, <http://www.apple.com/macosx/whats-new/gestures.html>.
 - [34] F. Wang and X. Ren, “Empirical evaluation for finger input properties in multi-touch interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1063–1072, April 2009.
 - [35] TUIO, TUIO, 2012, <http://www.tuio.org/>.
 - [36] Fajran, Tongseng TUIO event wrapper, 2012, <https://github.com/fajran/tongseng>.
 - [37] O. Chapuis, A. Bezerianos, and S. Frantzeskakis, “Smarties: an input system for wall display development,” in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*, pp. 2763–2772, May 2014.
 - [38] S. M. Peck, C. North, and D. Bowman, “A multiscale interaction technique for large, high-resolution displays,” in *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '09)*, pp. 31–38, Lafayette, La, USA, March 2009.
 - [39] C. Liu, O. Chapuis, M. Beaudouin-Lafon, E. Lecolinet, and W. Mackay, “Effects of display size and navigation type on a classification task,” in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*, pp. 4147–4156, May 2014.