Proxemic Cursor Interactions for Touchless Widget Control

Kieran Waugh
University of Glasgow
Glasgow, Scotland
k.waugh.1@research.gla.ac.uk

Mark McGill
University of Glasgow
Glasgow, Scotland
Mark.McGill@glasgow.ac.uk

Euan Freeman
University of Glasgow
Glasgow, Scotland
euan.freeman@glasgow.ac.uk

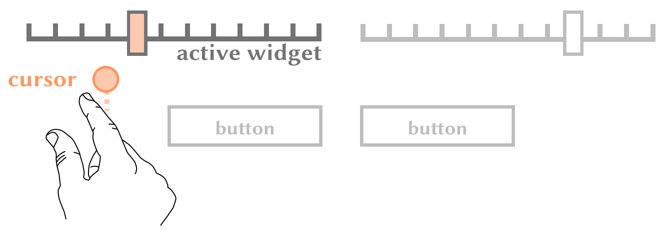


Figure 1: Proxemic cursor widgets allow users to activate the nearest element to a gesture-controlled cursor in a touchless user interface, without directly targeting it. This reduces input time and can improve the ergonomics of touchless input.

ABSTRACT

Touchless gesture interfaces often use cursor-based interactions, where widgets are targeted by a movable cursor and activated with a mid-air gesture (e.g., push or Pinch). Continuous interactions like slider manipulation can be challenging in mid-air because users need to precisely target widgets and then maintain an 'activated' state whilst moving the cursor. We investigated proxemic cursor interactions as a novel alternative, where cursor proximity allows users to acquire and keep control of user interface widgets without precisely targeting them. Users took advantage of proxemic targeting, though gravitated towards widgets when negotiating the boundaries between multiple elements. This allowed users to gain control more quickly than with non-proxemic behaviour, and made it easier to move between user interface elements. We find that proxemic cursor interactions can improve the usability of touchless user interfaces, especially for slider interactions, paving the way to more comfortable and efficient use of touchless displays.

CCS CONCEPTS

• **Human-centered computing → Gestural input**.

KEYWORDS

Mid-Air Gestures, Pinch Gestures, Proxemics, Touchless Interaction, Touchless Widgets

ACM Reference Format:
Kieran Waugh, Mark McGill, and Euan Freeman. 2023. Proxemic Cursor Interactions for Touchless Widget Control. In *The 2023 ACM Symposium on Spatial User Interaction (SUI '23), October 13–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3607822.3614525

SUI '23, October 13–15, 2023, Sydney, NSW, Australia

1 INTRODUCTION

Touchless gesture interfaces enable non-contact input to public displays and other computing devices through mid-air gestures, i.e., meaningful hand movements and poses. There is increased interest in touchless gesture technologies since they offer a more hygienic alternative to touchscreens, which is especially compelling due to growing concerns over the cleanliness of public displays [8, 15, 26, 29]. Additionally, touchless interaction enables input to out-of-reach displays [28] and is often preferred over touchscreen use when it reduces physical effort [24]. Commercial touchless software platforms have emerged recently, to add touchless input capabilities to existing touchscreen systems, e.g., Intel RealSense Touchless [23] and Ultraleap TouchFree [34]. Such platforms are being used in public settings like restaurants [36], retail [32, 37], and tourism [33].

Touchless interfaces often use cursor-based interactions, where users control an on-screen cursor via hand movements. Interface elements (like buttons and sliders) are then activated typically by an "AirPush" gesture (i.e., pushing their hand towards the display) [35]. However, AirPush gestures are prone to usability issues, e.g., uncertainty about how hand depth affects the cursor and target slips due to the Heisenberg effect [43], i.e., the forward motion of a finger can inadvertently alter the intended pointing position, thus complicating accurate selection. Continuous interactions like slider control are especially challenging here because users need to maintain that uncertain activated state whilst controlling the cursor [40]. Unlike direct touch input, touchless input provides feedback about how the system is responding to their cursor movements *before* they activate targeted widgets. Exploring alternative interaction techniques that minimise the requirement for precise targeting before widget activation could enhance the usability of touchless input, particularly for continuous cursor operations like slider control.

This paper investigates **touchless proxemic cursor interactions**, where user interface widgets are implicitly targeted when closest to the cursor. The nearest widget is automatically selected (e.g., Figure 1) and can be activated by gestures like finger Pinch or AirPush. Though not yet implemented in touchless public displays, proxemic cursor interactions have been found to benefit other input modalities like pointing devices [16] and can potentially address usability issues related to cursor instability and fatigue, make distant widgets more accessible, and allow interaction within a more comfortable range of human motion, especially for continuous operations like slider control [2, 22].

We present two experiments investigating touchless interaction with a proxemic cursor, to better understand the effects of proxemics on input behaviour. Users were tasked with making selections using sliders and buttons, and their performance was analysed based on time, accuracy, and cursor-targeting behaviour. The studies aimed to understand how people use touchless proxemic cursors and how they manage the boundaries between adjacent widgets, as this would give insight into how to use proxemic cursor interactions effectively. We also compared two activation gestures – AirPush and Pinch – to see how these affected proxemic cursor use.

We contribute a formative exploration of a proxemic cursor, particularly in the context of slider manipulation and multi-widget interactions. Our results provide evidence that touchless proxemic cursors significantly enhance touchless interactions making control acquisition faster and easier improving our capability to interact with touchless displays.

2 BACKGROUND

2.1 Pinch Gestures for Continuous Interactions

Touchless interfaces present several challenges, including uncertainty about interaction locations [12, 13], diminished feelings of control [7], and ambiguity about interaction states due to limited sensory feedback [11]. Despite these issues, button pressing is broadly adopted in touchless interfaces. Recent studies [40, 44] are starting to investigate more complex continuous interactions, such as touchless sliding for slider control and content scrolling, which have not been as extensively researched as discrete interactions. Continuous interactions in touchless interfaces, like maintaining widget control, are more complex than discrete actions like button pressing. Pinch gestures, which have two defined states and do not require hand displacement, have been introduced to ease continuous control [40, 44]. This approach enhances stability and simplifies interaction, which along with their reliability, has led to their adoption in mixed reality inputs [31], such as Microsoft HoloLens, Meta/Oculus Quest, and Apple Vision Pro.

Pinching not only serves as an alternative activation mechanism but also minimises the need for precise widget targeting in touchless interfaces. Recent research [40] used pinching as a universal control gesture for sliders, allowing users to control a slider from anywhere within the tracking range. This led to quicker slider control acquisition and improved target times compared to traditional methods. Additionally, it provides a simpler option compared to more sophisticated gesture techniques that reduce the need for precise targeting, e.g., summoning widgets to the cursor [18] or using cursor-less temporal correlation gestures [5, 12, 27, 38].

While the 'universal Pinch' gesture offers benefits, its limitation lies in its design where it is intended for use with a single user interface widget. To mitigate this and explore alternative activation gestures, this paper investigates *cursor proximity* for widget selection, allowing users to activate or control widgets without further cursor movement. These are termed **proxemic cursor widgets**, drawing inspiration from other techniques where cursor proximity enhances targeting.

2.2 Cursor Proximity Interactions

Cursor proximity has been leveraged across various input modalities to enhance pointing and targeting tasks. Some conventional methods, like using a mouse or trackpad, dynamically resize targets or apply lens distortions to enlarge targets near the cursor [19]. Alternatively, the cursor itself can be resized to 'snap' to the closest target, as seen in Bubble Cursor [16] and its adaptations for mid-air [9] and ray-cast pointing [3]. In displays where user interface elements cannot be resized (e.g., tangible objects), 'virtual targets' can be used to allow cursors to more easily select the nearest target by targeting its surrounding area instead [14].

Several studies have investigated the combination of various input methods like gaze, touch and gesture for cursor positioning and refinement. Gaze-Touch [30] used eye tracking for cursor positioning and touch input for precise adjustments, while Gaze+Gesture [4] applied similar techniques for mid-air gestures, enabling precise inputs for smaller targets. The Gaze-Hand Alignment [25] study introduced novel menu selection techniques that use gaze and mid-air gestures, minimising hand movement at the start of an input operation. However, gaze tracking is not always feasible, such as with public displays situated 1-2 metres away. In our study, we concentrate on touchless gestures as the primary input mode, though our techniques are not limited to this modality.

These interaction techniques all use cursor proximity to enhance targeting by enlarging the target width and/or reducing the distance to the target. Proxemic cursor widgets streamline targeting by auto-selecting the nearest user interface widget, which can then be activated with an appropriate gesture like a Pinch or an AirPush. Another way of looking at this concept is that we are using cursor proximity to enable the same gesture(s) to be reused in a localised way, mapped to a single user interface widget. Other mid-air gesture interfaces have implemented this principle to reuse small gesture sets. For instance, buttons can be grouped so that one-handed count gestures can select numbered targets from the group nearest to the cursor [11]. In our case, cursor proximity identifies the target for an arbitrary activation gesture controlling a slider widget.

3 TOUCHLESS PROXEMIC CURSOR WIDGETS

Proxemic cursor widgets are touchless user interface controls that auto-select when closest to the cursor (mapped in this case to hand position) and trigger with a suitable activation gesture. Unlike typical touchless widgets, the closest widget to the cursor is targeted automatically, enabling activation gestures *without* necessitating precise cursor positioning (as in Figure 1). Activation gestures (e.g., Pinch and AirPush) trigger the selected control, providing a clear state change. In discrete interactions like button pressing, these gestures trigger input events, while in continuous interactions like slider control, they act as a mode switch to control and move the slider, released when the gesture ends.

Identifying the nearest widget to the cursor is relatively simple for basic controls like buttons by calculating the minimum distance between the cursor and the closest pixel from each control's bounding box. For more intricate controls with multiple parts, such as sliders consisting of a slider bar and a movable handle, determining proximity is more complex. In our proxemic sliders, we determined slider proximity using the current handle position. Although using the entire slider's bounding rectangle could be an alternative, pilot testing with a small group of interaction designers found that the handle position better aligned with users' perception of how neighbouring sliders should react to cursor movement, especially when handles are at the slider bar ends. By using these rules, a touchless user interface can automatically target a suitable widget.

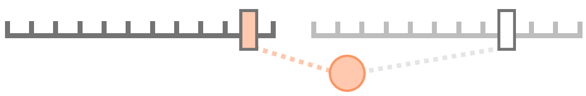


Figure 2: We define the nearest slider as the minimum distance to the slider handle. In this case, the leftmost slider would be targeted.

Upon initiating an AirPush or Pinch gesture, the targeted widget activates: buttons are 'pressed' on release, like a touchscreen tap or mouse click; for sliders, the user gains control of the handle and can adjust its position with corresponding hand movements.

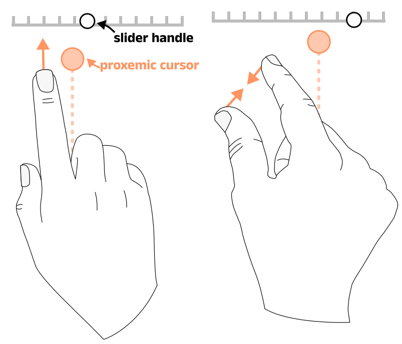
Proxemic cursor widgets enhance interaction by reducing the need for precise targeting in touchless cursors by adapting a 'universal' Pinch gesture to handle multiple widgets. While the cursor is no longer a precise pointer, it provides feedback on hand position and aids in targeting. While our study did not adjust cursor appearance or render target boundaries to establish a baseline evaluation, these factors can be explored in future work.

This paper examines proxemic cursor widgets, specifically for touchless slider input and button activation as these user interface components are widely used and enable us to look at both continuous and discrete interactions with proxemic cursors. Enhancing the efficiency of touchless interactions, particularly in the context of slider widgets, is of significant importance. This type of interaction interface, despite its ubiquitous nature, often presents substantial challenges with respect to targeting accuracy. We present two experiments comparing performance in touchless selection tasks using proxemic and non-proxemic cursors.

4 EXPERIMENTAL DESIGN

We conducted two within-subjects user studies to evaluate the effects of a proxemic cursor on touchless input. As highlighted in section 2, we see issues with continuous interactions and maintaining control over movable widgets. Therefore, we first test slider input tasks to understand how the proxemic cursor performs on solely continuous input. Once the efficacy was confirmed on a single widget type (sliders), we then examined whether this would hold across other widget types. Across both studies, our main aim was to to compare the impact of proxemic cursor behaviour against traditional touchless cursor behaviour during input tasks. We hypothesised proxemic cursor behaviour would enhance task performance due to reduced cursor movement time and effort. However, we want to understand *how* people take advantage of proxemic cursor behaviour, such as the proximity of cursor movement before interaction, how users manage widget boundaries, and if they adapt to proxemic targeting or stick with conventional behaviours.

We also explored how different activation gestures, namely *AirPush* and *Pinch* (see Figure 3), impact usability and task performance. Although mid-air Pinch gestures have shown potential benefits [40, 44], we questioned whether cursor proximity might equalise performance by preventing target slips during AirPush gestures. For cursor movement, we used an absolute mapping of hand position to cursor position to align the centre of the sensing space with the display's centre.



(a) AirPush **(b) Pinch**

Figure 3: Activation gestures: AirPush (a), activated after a 100mm push forward and Pinch (b), activated once the thumb and index finger are 20mm apart.

Our touchless slider interface was shown on a 27-inch display placed atop a table, with an Ultraleap Stereo IR 170 sensor centred beneath the display, facing upwards to track hands in the space in front of the screen. Users were positioned approx. 1 m from the screen. A large button was placed on the table within reach of the non-dominant hand, and was used by participants to signal the end of each task whilst gesturing with their dominant hand. To begin, participants were introduced to the four interaction techniques and three slider layouts, completing practice tasks for each combination until they were comfortable. Participants were instructed to complete the tasks as quickly and as accurately as possible and started with their hands in front of their body. Timing started from when the dominant hand was extended above the sensor to when the participant pressed a button with their other hand, ensuring consistent timing. Tasks were presented in blocks for each condition in a counter-balanced order using a Latin square.

4.1 Measurements

For each task, we recorded overall **task time** , **time to acquire the first widget** ($T_{acquire}$), **time to transition**($T_{transition}$) between widgets, and time to initially reach the target (**time to target** (T_{target})). Figure 4 shows how these measurements relate to key interaction events. This division allowed us to understand how each interaction technique and slider layout influenced different interaction phases. The two periods not covered by these measurements relate to user adjustments after overshooting or releasing the handle, as our focus was primarily on slider acquisition and initial movement. We also measured the **handle distance** between the final handle position and the exact intended target position (recorded in pixels and converted to mm).

We tracked **cursor position** from the start to end of each slider interaction to derive the **proxemic cursor distance** at the beginning of each interaction. After each condition block, we used the NASA Task Load Index (**TLX**) survey [20] to measure task workload. The experiment concluded with a semi-structured interview, starting with participants ranking their preferred interaction techniques, which served as a discussion prompt about their experience. Our experiment data is available via the Zenodo open research data platform [41].

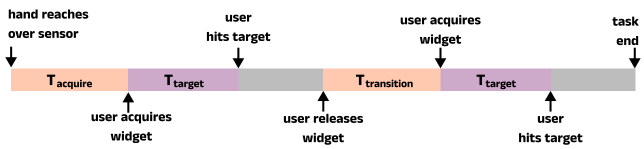


Figure 4: This timeline shows how task timing measures are calculated from interaction events where T denotes time.

5 EXPERIMENT 1: MULTIPLE SLIDER INPUTS

5.1 Method

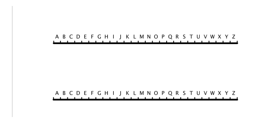
Our aim in this experiment was to investigate how relative slider layout affected proxemic cursor behaviour, as this might influence the types of cursor movement necessary for input. Participants were asked to select two alphabetic letters using two adjacent sliders. The slider bars allowed alphabetical selection from A–Z, e.g., as might be used when filtering a collection of items, destinations, songs, etc. by moving the slider handles to designated characters from the starting position. The order of sliders was altered to observe any changes in user behaviour when crossing widget boundaries in different directions, i.e., from Slider 1 to Slider 2 and vice versa. We varied the position and orientation of the sliders on screen, using three slider layouts: **HSS** (horizontal side-by-side), **HTB** (horizontal top and bottom), and **VSS** (vertical side by side) These layouts provide variety in slider orientation (horizontal vs vertical) and target size (e.g., because the HTB layout allows wider slider bars), giving some insight into the effect of user interface layout on touchless slider usability. The absence of a vertical top and bottom (VTB) condition was due to the landscape screen orientation rendering the target size too small. Our choice of two sliders was the minimum to necessitate proxemic cursor behaviour without over complicating the initial study. Figure 5 shows screenshots of each slider layout.

We studied three independent variables: *Cursor type* (Proxemic, Non-Proxemic), *Gesture type* (AirPush, Pinch), and *Slider Layout* (HSS, HTB, VSS), giving 12 conditions. The sequence was counterbalanced using Latin squares, and each participant completed twelve trials per condition, with a balanced mix of starting sliders. All tasks began with the sliders at 'A' for consistency. Slider targets were chosen beyond the first three notches (A-C) to ensure a distinct start and target position.

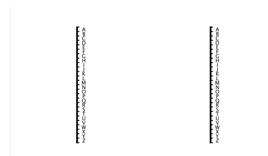
5.2 Participants

We recruited 21 participants (14 male, 6 female, 1 other; average age 29 years; 18 right-handed, 2 left, 1 ambidextrous) via institution mail lists and posters. Participants were compensated with £10.

(a) HSS (horizontal side-by-side)



(b) HTB (horizontal top-and-bottom)



(c) VSS (vertical side-by-side)

Figure 5: Screenshots of the three slider layouts. Each slider has 26 regions, labelled from A to Z.

5.3 Results

All time measures did not have a normal distribution (via Shapiro-Wilk test) and so the Aligned-Rank Transform [42] was used prior to parametric statistical analysis, with post hoc comparisons using the ART-C method [10]. Two blocks of data (from a total of 252) were omitted from the analysis because the log data was corrupted.

5.3.1 Handle Acquisition Distance. Mean distance between the slider handle and the intended target position was 3.3 mm (SD 1.1 mm); for context, each slider target width varied from 11–17.9 mm depending on layout. Figure 6 (a) shows the mean handle distance for each condition. A repeated-measures t-test found no significant difference between the handle distance for the first and second sliders ($t(2326) = 1.29$, $p = .2$). A repeated-measures ANOVA found significant main effects of **Gesture** ($F(1, 218) = 10.25$, $p = .002$) and **Layout** ($F(2, 218) = 7.29$, $p < .001$). There was no main effect of **Cursor** ($F(1, 128) = .22$, $p = .64$) and there were no interaction effects between any factors (all $F \leq .87$, $p \geq .42$).

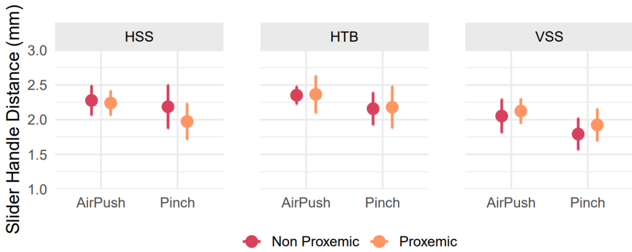
Post hoc contrasts for **Gesture** found significantly larger distances for AirPush than Pinch ($t(218) = 3.2$, $p = .002$). Post hoc contrasts for **Layout** found significantly larger distances for both horizontal sliders than for the vertical slider layout (both $t(218) \geq 2.7$, $p \leq .02$). There was no sig. difference between horizontal layouts ($t(218) = 1.02$, $p = .56$).

5.3.2 Task Time. Mean task time for both selections was 13049 ms (SD 2953 ms). Figure 6 (b) shows the mean task time for each condition. A repeated-measures ANOVA found significant main effects of **Cursor** ($F(1, 218) = 17.32$, $p < .001$), **Gesture** ($F(1, 218) = 39.09$, $p < .001$), and **Layout** ($F(2, 218) = 4.54$, $p = .01$). There was a significant interaction effect for **Cursor** × **Layout** ($F(2, 218) = 3.52$, $p = .03$). No other interactions were significant (all $F \leq 3.84$, $p \geq .05$).

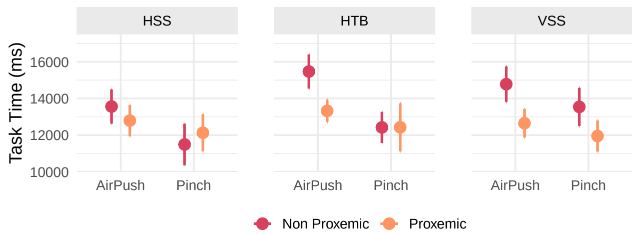
Post hoc contrasts for **Cursor** found significantly shorter task times for Proxemic than Non-Proxemic cursors ($t(218) = 4.16$, $p < .001$).

Post hoc contrasts for **Gesture** found significantly shorter task times for the Pinch than AirPush ($t(218) = 6.52$, $p < .001$).

Post hoc contrasts for slider **Layout** found significantly shorter task times for the HSS than HTB layout ($t(218) = 2.82$, $p = .01$); the

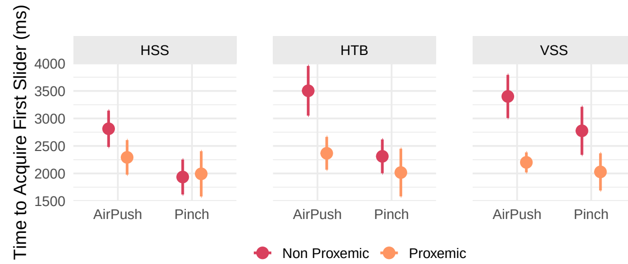


(a) Slider Handle Distance

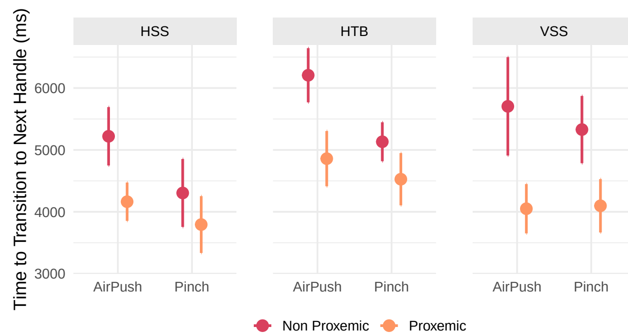


(b) Task Time

Figure 6: Mean slider handle distance (a) and task time (b) for each condition. Error bars show 95% CIs.



(a) Time to Acquire First Handle



(b) Time to Transition to Second Handle

Figure 7: Mean time to acquire first (a) and second (b) slider handles for each condition. Error bars show 95% CIs.

other layout comparisons were not significantly different (both $t(218) \leq 2.34$, $p \geq .052$).

Post hoc contrasts for the **Cursor** × **Layout** interaction effect revealed that Proxemic cursors were faster than Non-Proxemic cursors for HTB and VSS layouts ($t(218) \geq 3.17$, $p \leq .02$) but not for HSS ($t(218) = .27$, $p = .99$).

5.3.3 Time to Acquire First Slider ($T_{acquire}$). Mean time to acquire control of the first slider handle (i.e., for the first phase of the task) was 2473 ms (SD 1204 ms). Figure 7 (a) shows the mean time for each condition. A repeated-measures ANOVA found significant main effects of **Cursor** ($F(1, 218) = 77.37$, $p < .001$), **Gesture** ($F(1, 218) = 61.37$, $p < .001$) and **Layout** ($F(2, 218) = 7.57$, $p < .001$),. There were significant interaction effects for **Cursor** × **Gesture** ($F(1, 218) = 8.92$, $p = .003$) and **Cursor** × **Layout** ($F(2, 218) = 4.51$, $p = .01$). No other interactions were significant (both $F \leq .73$, $p \geq .48$).

Post hoc contrasts for **Cursor** found it took significantly less time to acquire control of the first slider handle when using a Proxemic cursor than Non-Proxemic cursor ($t(218) = 8.80$, $p < .001$).

Post hoc contrasts for **Gesture** found it took significantly less time to acquire control of the first slider handle when using the Pinch activation gesture ($t(218) = 7.83$, $p < .001$).

Post hoc contrasts for **Layout** found it took significantly less time to acquire the slider handle for the HSS layout than for the others (both $t(218) \geq 2.58$, $p \leq .03$). There was no sig. difference between HTB and VSS ($t(218) = 1.22$, $p = .44$).

Post hoc contrasts for the **Cursor** × **Gesture** interaction effect found no significant difference between AirPush × Proxemic and Pinch × No-Proxemic ($t(218) = .58$, $p = .94$). All other pairwise combinations were significantly different, with Proxemic gestures being faster than Non-Proxemic gestures (all $t(218) \geq 3.37$, $p \leq .005$).

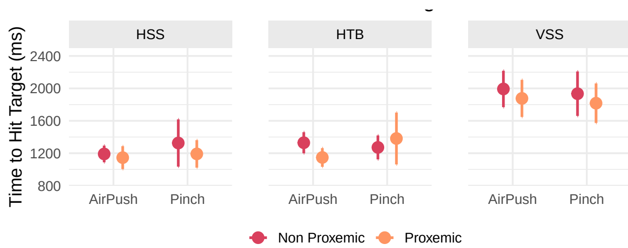
Post hoc contrasts for the **Cursor** × **Layout** interaction effect found that HSS layout was the only layout where there was no significant difference between Proxemic and Non-Proxemic cursors ($t(218) = 2.68$, $p = .08$). In all other layout comparisons, Proxemic cursors took less time to acquire (both $t(218) \geq 6.04$, $p \leq .001$).

5.3.4 Time to Move Between Sliders ($T_{transition}$). Mean time to acquire the second slider (after ending the first slider interaction) was 1840 ms (SD 671 ms). Figure 7 (b) shows the mean time for each condition. A repeated measures ANOVA found there were significant main effects of **Cursor** ($F(1, 218) = 185.08$, $p < .001$) and **Gesture** ($F(1,218) = 18.59$, $p < .001$). There was no significant main effect of **Layout** ($F(2, 218) = 2.49$, p =,.09) and there were no significant interaction effects (all $F \leq 1.73$, $p \geq .19$).

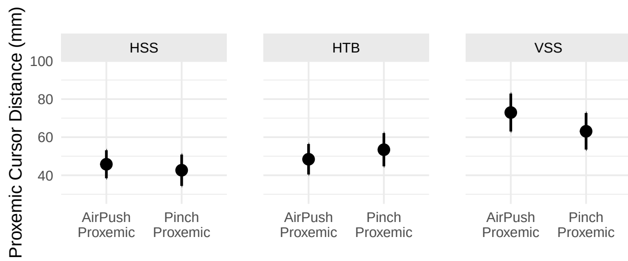
Post hoc contrasts for **Cursor** found it took significantly less time to acquire control of the second slider handle when using Proxemic cursors ($t(218) = 13.6$, $p < .001$).

Post hoc contrasts for **Gesture** found it took significantly less time to acquire control of the second slider handle when using the Pinch gesture than AirPush ($t(218) = 4.31$, $p < .001$).

5.3.5 Time to Hit Slider Target (T_{target}). Mean time for the slider handle to first reach the target position was 1469 ms (SD 615 ms). Figure 8 (a) shows the mean time to target for each gesture and slider layout. A repeated-measures t-test found no significant difference between the first and second interactions ($t(2326) = .99$, $p = .32$). A repeated-measures ANOVA found significant main effects of **Cursor** ($F(1, 218) = 5.68$, $p = .02$) and **Layout** ($F(2, 218) = 71.4$, $p < .001$) on target time. There was no main effect of **Gesture** ($F(1, 218) = 1.34$, $p = .25$), and no significant interaction effects (all $F \leq 1.10$, $p \geq .34$).



(a) Time to Reach Target



(b) Cursor Distance

Figure 8: Mean time to first reach target (a) and proxemic cursor distance (b). Error bars show 95% CIs.

Post hoc contrasts for **Cursor** found it took significantly less time to reach the target when using Proxemic cursors ($t(218) = 2.38$, $p = .02$): 1429 ms vs 1508 ms for proxemic and non-proxemic.

Post hoc contrasts for slider **Layout** found it took significantly longer to first reach the target on the vertical sliders than for both horizontal layouts (both $t(218) \geq 9.82$, $p < .001$). There was no significant difference between horizontal layouts ($t(218) = .97$, $p = .6$).

5.3.6 Proxemic Cursor Distance to Slider. We measured cursor position at the beginning of each slider interaction. For the Non-Proxemic conditions, the cursor needed to overlap the slider handle to take control. For the Proxemic conditions, however, users could take control of a slider handle when it was the nearest to the cursor. The mean proxemic cursor distance was 54.5 mm (SD 31.3 mm), i.e., there were 54.5 mm between the cursor and slider handle at the point when the user performed the gesture to take control of the handle. This was over four times the distance necessary for the Non-Proxemic conditions. Figure 8 (b) shows the mean proxemic cursor distance for each gesture and layout.

A repeated-measures ANOVA found a significant main effect of slider **Layout** on cursor distance ($F(2, 98) = 17.74$, $p < .001$); there was no main effect of **Gesture** ($F(1, 98) < .001$, $p = .98$). Post hoc contrasts for **Layout** found significantly lower proxemic cursor distances for the horizontal than vertical slider layouts (both $t(98) \geq 3.99$, $p < .001$). There was no significant difference between the horizontal slider layouts ($t(98) = 1.82$, $p = .17$).

To illustrate the above effects, we also plotted the cursor position at the start of each slider interaction to see where the cursor was positioned relative to the slider handle, when the user took control. Figure 9 shows the distribution of cursor positions for each gesture and slider layout combination. This shows the extent to which users took advantage of the proxemic cursor behaviour and suggests how they negotiated the boundaries between proxemic activation zones.

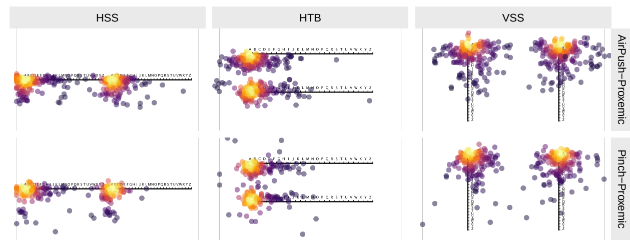
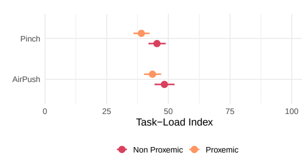
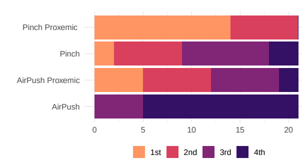


Figure 9: Cursor positions when the user took control of the slider handles with brighter colours indicating greater density of cursor positions. Only the proxemic gestures are shown, as the non-proxemic gestures require the cursor to overlap the slider handle (i.e., cursor distance ≤ handle radius). A full-resolution version is available in the supplementary material.

5.3.7 Task-Load Index. Overall TLX score for each gesture was computed from the NASA-TLX survey responses using the 'raw TLX' method [20]. Mean overall TLX score was 44.1 out of 100 (SD 16.6), as shown in Figure 10 (a). We applied the Aligned-Rank Transform to the TLX scores so we could perform multi-factor analysis by **Cursor** and **Gesture**. A repeated-measures ANOVA found significant main effects of **Cursor** ($F(1, 60) = 8.49$, $p = .005$) and **Gesture** ($F(1, 60) = 4.36$, $p = .04$). There was no significant interaction between these factors ($F(1, 60) = .036$, $p = .85$).



(a) Task-Load Index



(b) Preference Ranks

Figure 10: Mean task-load index for each gesture (a) and distribution of rankings (b). Error bars show 95% CIs.

Post hoc contrasts for **Cursor** confirmed that TLX scores were lower for Proxemic than Non-Proxemic interaction techniques ($t(60) = 2.91$, $p = .005$). Post hoc contrasts for **Gesture** confirmed that TLX scores were lower for Pinch than AirPush interaction techniques ($t(60) = 2.09$, $p = .04$).

5.3.8 Preference Ranks. Participants ranked the four interaction techniques in order of preference. Figure 10 (b) shows a count of ranks for each technique. Pinch × Proxemic was the most preferred technique with 14 people ranking it as their most preferred. Friedman's test found a significant effect of interaction technique on rank: $\chi^2 = 37.97$, $p < .001$. Post hoc Nemenyi comparisons found that AirPush was significantly less preferred than all others (all $p \leq .021$), and Pinch × Proxemic was significantly more preferred than Pinch (p = .007); no other comparisons were sig. different (p ≥ .079).

6 EXPERIMENT 2: SLIDERS AND BUTTONS

The aim of the second experiment was to investigate how people used proxemic cursors to interact with different kinds of widgets (continuous input using sliders, discrete input using buttons) and to explore how they navigated between a larger number of interface elements. We evaluated proxemic cursor behaviour with both slider and button widgets, with an interactive slider positioned above three side-by-side buttons. In contrast to the previous experiment where users just moved between two slider widgets in a variety of layouts, this experiment gave users narrower effective target widths and required horizontal as well as vertical targeting motions. We also controlled the horizontal and vertical spacing between widgets to see how that affected proxemic cursor behaviours.

6.1 Method

Participants were asked to make a slider selection and then activate one of three buttons (or vice versa). The slider bar allowed an alphabetical selection from A–Z while the three buttons were labelled numerically from 1–3. Selection order was altered to observe changes in user behaviour when crossing widget boundaries in different directions, i.e., from Slider to Button and vice versa. We tested different values for widget separation (64px, 128px and 256px) as it has been noted that a touchless target separation of under 64px dramatically increases selection difficulty [17]. We examined all combinations of the separation values in order to determine if the vertical or horizontal spacing had an effect on proxemic cursor use. Figure 11 shows an example layout.

There were three independent variables in this within-subjects design: Gesture type (AirPush, Pinch), Vertical separation (64px, 128px, 256px) and Horizontal separation (64px, 128px, 256px), giving 18 conditions. Tasks were presented in blocks for each condition, in a counter-balanced order using a Latin square design. Each participant completed 12 trials per condition. All tasks started with the slider at 'A'. Slider targets were chosen beyond the first three notches (A-C) to ensure a distinct start and target position.

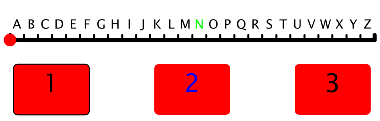


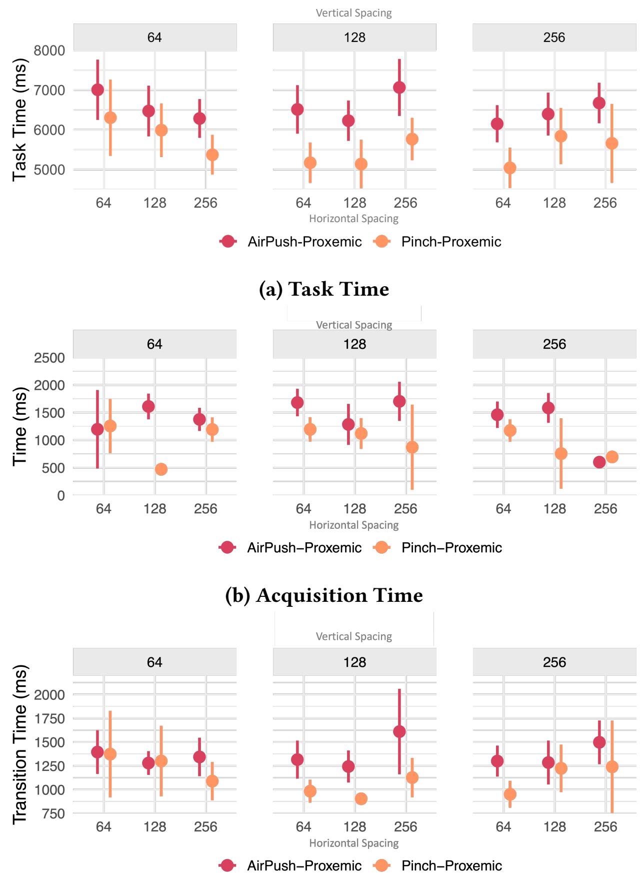
Figure 11: Experiment task screenshot. There are 26 slider notches (A–Z) and three buttons (1–3). Shows a vertical and horizontal separation of 64px and 256px, respectively.

6.2 Participants

We recruited 20 participants (14 male, 6 female, average age 28 years, 18 right-handed, 1 left-handed, 1 ambidextrous) via institution mailing lists and posters. The hour-long study, approved by our institution ethics committee, paid participants £10.

6.3 Results

All time measures did not have a normal distribution (via Shapiro-Wilk test) and so the Aligned-Rank Transform [42] was used prior to parametric statistical analysis, with post hoc comparisons conducted using the recent ART-C method [10].



(a) Task Time

(b) Acquisition Time

(c) Transition Time

Figure 12: Task Time(a), Acquisition Time (b) and Transition Time(c) for each condition. Error bars show 95% CIs.

6.3.1 *Task Time.*

The mean task time for both selections was 6057 ms (SD 2061 ms). Figure 12 (a) shows the mean task time for each condition. A repeated-measures ANOVA found significant main effects of **Gesture** ($F(1, 323) = 46.86$, $p < .001$) and the interaction of **Horizontal** × **Vertical** spacing ($F(4, 323) = 3.93$, $p = .003$). No other interactions were significant (all $F \leq 0.31$, $p \geq .31$).

Post hoc contrasts for **Gesture** found significantly shorter task times for Pinch versus AirPush ($t(323) = 6.85$, $p < .0001$).

Post hoc contrasts for the **Horizontal** × **Vertical** interaction found no significant differences (all $T \leq 2.73$, $p \geq .09$).

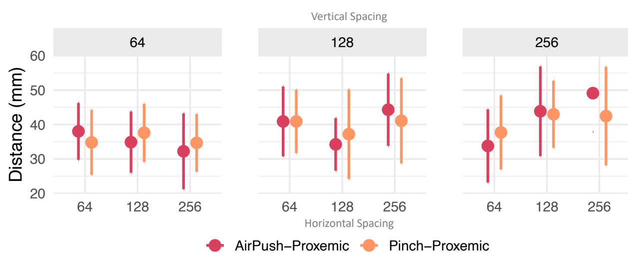
6.3.2 *Time to Acquire First Widget ($T_{acquire}$).*

Mean time to acquire control of the first widget (i.e., for the first phase of the task) was 1178 ms (SD 1461 ms). Figure 12 (b) shows the mean time for each condition. A repeated-measures ANOVA found significant main effect of **Gesture** (F(323) = 36.84, p < .001). No other interactions were significant (F ≤ .18, p ≥ .28).

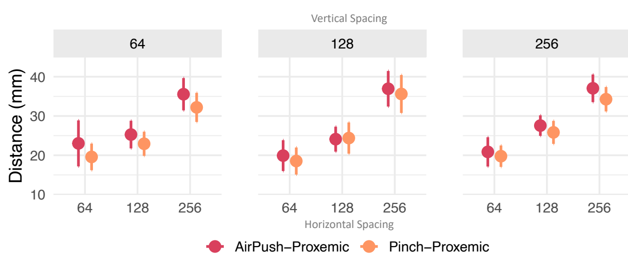
Post hoc contrasts for **Gesture** found it took significantly less time to acquire control of the first widget when using the Pinch activation gesture (t(323) = 52.1, p < .0001).

6.3.3 Time to Move Between Widgets ($T_{transition}$). The mean time to acquire the second widget (after ending the first widget interaction) was 1247 ms (SD 673 ms) Figure 12 (c) shows the mean transition time for each condition. A repeated measures ANOVA found there were significant main effects of **Gesture** (F(1, 323) = 33.53, p < .001). No other interactions were significant (all F ≤ 0.57, p ≥ .15).

Post hoc contrasts for **Gesture** found significantly shorter transition times for Pinch versus AirPush (t(323) = 49.9, p < .0001).



(a) Slider Activation Distance



(b) Button Activation Distance

Figure 13: Slider Activation Distance (a) and Button Activation Distance (b) for each condition. Error bars show 95% CIs.

6.3.4 Slider Handle Distance. The mean distance between the slider handle and the intended target position was 1.64 mm (SD 0.84 mm); for context, the slider target width was 14 mm. A repeated-measures ANOVA found significant main effects of **Gesture** (F(1, 323) = 53.21, p < .001). No other interactions were significant (all F ≤ 1.45, p ≥ .23).

Post hoc contrasts for **Gesture** found a significantly shorter handle distance from the target for the Pinch than AirPush activation gesture (t(323) = 63.8, p < .0001).

6.3.5 Slider Activation Distance. The mean distance when activating the slider was 38.93 mm (SD 25.03 mm). Figure 13 (a) shows the mean task time for each condition. A repeated-measures ANOVA found no significant main effects (all F ≤ 0.07, p ≥ .12).

6.3.6 Button Activation Distance. 60% of button selections were made within the button boundaries. When activating **outwith the boundaries**, the mean distance was 26.86 mm (SD 10.65 mm). Figure 13 (b) shows the mean button activation distance for each condition. A repeated measures ANOVA found there were significant main effects of **Gesture** (F(1, 323) = 4.48, p = .03), **Vertical Spacing** (F(2, 323) = 3.82, p = .02), and **Horizontal Spacing** (F(2, 323) = 181.22, p < .001). No other interactions were significant (all F ≤ 0.89, p ≥ .46).

Post hoc contrasts for **Gesture** found significantly further activation distance for the AirPush than Pinch activation gesture (t(323) = 22.5, p = .04).

Post hoc contrasts for **Horizontal Spacing** found significantly further activation distance for 64px versus 128px (t(323) = 70.6, p < .0001), 64px versus 256px (t(323) = 176.1, p < .0001), and 128px versus 256px (t(323) = 105.6, p < .0001).

Post hoc contrasts for **Vertical Spacing** found significantly further activation distance for 64px versus 128px (t(323) = 32.5, p = .03).

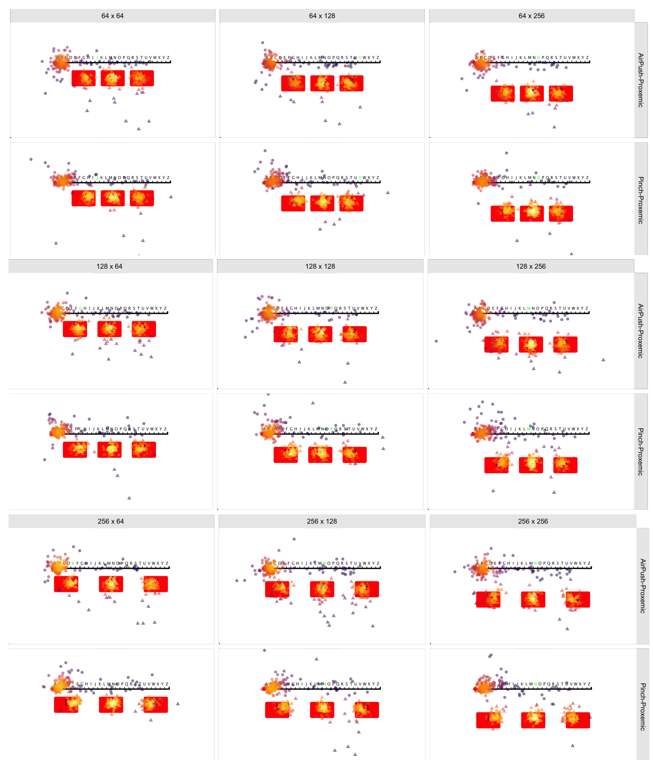


Figure 14: Cursor positions at the time of widget activation, with brighter colours marking higher density areas. Slider and button activations are marked by circles and triangles, respectively. A full-resolution version is available in the supplementary material.

6.3.7 Task-Load Index. Overall TLX score for each gesture was computed from the NASA-TLX survey responses using the 'raw TLX' method [20]. Mean overall TLX score was 32 out of 100 (SD 15), as shown in Figure 15. We applied the Aligned-Rank Transform to the TLX scores so we could perform multi-factor analysis by **Gesture**, **Horizontal Spacing** and **Vertical Spacing**. A repeated-measures ANOVA found significant main effects of **Gesture** (F(1,

319) = 16, p < .001) and **Horizontal Layout** (F(2, 319) = 3.24, p = .004). No other interactions were significant (all F ≤ 1.53, p ≥ .19).

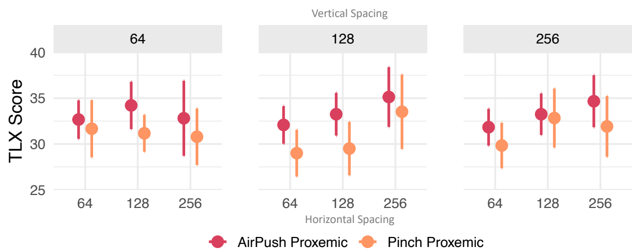


Figure 15: Mean task-load index for each condition.

Post hoc contrasts for **Gesture** showed that TLX scores were lower for Pinch than AirPush (t(319) = 19.4, p < .0001).

Post hoc contrasts for **Horizontal Spacing** showed that TLX scores were lower for 64 px than 256 px spacing (t(319) = 14.9, p = .03).

7 DISCUSSION

7.1 Proxemic Cursor Usability

Proxemic cursor interaction enabled users to be less precise when taking control of the slider handles, leading to faster acquisition times. Users noted they liked not having to aim so precisely as it made it easier to provide input. Despite reduced precision, there was no detrimental effect on slider or button selection accuracy. In experiment 1, users took control over the slider handles from a mean proxemic cursor distance of 54.5 mm, but the final handle distance relative to the target position was not significantly different for proxemic and non-proxemic cursors (2.1 mm, 95% CI [2.02, 2.25] mm vs 2.1 mm, 95% CI [2.01, 2.26] mm, p = .64). This suggests imprecise targeting of the handle was not also encouraging imprecise selection from the slider, or having a negative effect on selection trajectory because of, e.g., gesture sensing issues. In experiment 2, we also see accurate slider selection with a final handle distance relative to the target position of 1.64 mm despite users acquiring control of the slider handles from a mean proxemic cursor distance of 38.9 mm.

Proxemic cursor interactions may reduce input time by making it easier for users to begin providing input. Mean task times were 13 and 6 seconds, respectively, for both experiments, during which users completed two widget selections. To place these results into context, other touchless slider studies had a mean task completion time of 9.2 seconds for just one slider selection [40]. In experiment 1, task completion times were lower for the proxemic cursor conditions than non-proxemic (12.55s vs 13.54s, p < .001). This difference is largely explained by the reduced time to acquire control of the first slider (2.1s vs 2.8s, p < .001) and the reduced time to take control of the second slider (1.5s vs 2.2s, p < .001). Interview comments suggested that some users also perceived the proxemic cursor techniques as feeling faster, so were aware of the difference. In experiment 2, task time was approximately halved (6067 ms). As the task involved a single movable slider widget and discrete buttons, we expected this to be lower. However, we still see lower time to acquire control of the first widget (1.1s) and transition to the second widget (1.2s). The post-experiment interview comments suggested that the proxemic cursor allowed users to more easily gain control without having to target precisely.

Proxemic cursor interaction also improved the ergonomics of touchless slider interaction. Users often chose a more comfortable posture and did not need to move their hand as much for the cursor to be able to activate a slider handle. Similarly, users could use lower-effort cursor movement trajectories because there was less need for a precise aimed movement towards the slider handle. Together, these could lead to reduced joint tension, lowering fatigue during prolonged interactions [2, 22]. In experiment 1, eleven participants noted improved comfort from proxemic interactions during the interview, e.g., because they could keep their hand lower and did not need to move their hand as much. We further see this in experiment 2 where Figure 14 shows a distinct user behaviour pattern of selections towards the centre of the display. We see a large concentration of selections on the right side of button 1, the left side of button 3 and the centre of button 2. Post-experiment interviews highlighted that users again kept their hands lower and made smaller movements towards the target area, keeping the cursor in the centre of the display to reduce arm movements.

7.2 Targeting Behaviours

Although users took advantage of the proxemic cursor when using sliders, they often targeted buttons by moving the cursor directly over the button. When using the proxemic cursor, the mean proxemic cursor distance was 27 mm, though as seen from Figure 14, there is a greater density of selections over the buttons. Interview comments suggest the larger button surfaces allowed rapid selection with reduced need to rely on proxemic targeting. For smaller-sized buttons (comparable to the slider handle size) we anticipate increased use of proxemic targeting.

In experiment 1, we investigated different slider layouts. When using the non-proxemic cursor, users expressed more difficulty in acquisition and control with vertical sliders (VSS), with higher task workload ratings (46 vs 40, p < .001). Post-experiment interviews suggest that users benefit from the proxemic cursor here as it allowed them to perform smaller downward hand movements in a position that was more comfortable. However, there was no significant difference in the number of gesture activations between the cursor types for the VSS layout (3 vs 3, p = .49). Despite this we still see significantly faster task times (12.5s vs 13.5s, p < .001) along with faster times to reach the target (1.4s vs 1.5s, p = .02). When using vertical sliders for touchless input, we recommend placing no widgets directly beneath them so that users can take full advantage of proxemic targeting from a comfortable hand position.

In experiment 2, there was no significant difference in the proxemic cursor distance for the slider across all widget separation values. Post-experiment interviews also highlighted that users activated the slider without precise targeting regardless of the separation from the buttons, due to its position at the top of the screen. Users took advantage of the location by rapidly moving the cursor towards the handle and performing the gesture once highlighted. When activating outwith the button bounding box, predictably, we see significant increases in activation distance with increases in horizontal spacing (5.3 mm vs 7 mm vs 8 mm, p < .0001) but only see a significant difference in a large change in vertical spacing

64 px to 256 px (26 px vs 28 px, $p = .03$). Post-experiment interviews revealed that twelve participants found proxemic selections more difficult with the smaller horizontal spacing. These results suggest that despite the changes in widget separation values, the proxemic cursor allowed fast and accurate selections regardless of the spacing the user had available to navigate. We can also see this in Figure 14, where there is a distinct pattern (across all layouts) of selections below the slider handle, despite the proximity of the buttons to the handle. Therefore, we recommend the use of the proxemic cursor regardless of widget spacing. Future applications should also account for the possibility of diminishing returns when decreasing the proxemic zone and maintain intuitive boundaries [21].

7.3 Activation Gestures

We compared two activation gestures for proxemic cursor widgets: AirPush and Pinch. AirPush is a widely used activation gesture for touchless displays and pinching shows promise as an alternative because it is quick to perform [31] and has more clearly defined and easily recognisable (de)activation states [40, 44]. Interview comments from one participant [P4] suggested an additional benefit of Pinch was the implicit tactile feedback when their fingers touched: "*it was a good physical indicator of when I was selecting*", whereas the end point of AirPush was less obvious.

Pinch generally outperformed AirPush in our slider tasks: faster task times (12.3s vs 13.8s, $p < .001$) and (5.6s vs 6.6s, $p < .001$), faster acquisition times (2.2s vs 2.8s, $p < .001$) and (0.9s vs 1.3s, $p < .001$), faster transition times (1.7s vs 2.0s, $p < .001$) and (1.1s vs 1.4s, $p < .001$), and lower TLX scores (42 vs 46, $p = .04$) and (31 vs 33, $p < .0001$). However, once users were actually in control of the handle, there was only a very small difference in the time taken to first reach the target (1.49s vs 1.45s, $p = .25$) and (0.9s vs 1s, $p < .001$). These findings suggest the Pinch gesture leads to faster activation of touchless sliders. For button selections, we further see faster acquisition times for the Pinch gesture (0.9s vs 1.1s, $p < .0001$).

Although Pinch generally outperformed AirPush, we saw an increased preference for AirPush when using the proxemic cursor (compared to without). AirPush performed well for button activation in particular, with people taking advantage of the proxemic cursor more compared to Pinch, i.e., activating them from a further distance (25.9 mm vs 27.8 mm, $p = .04$). An interesting topic for future work would be to investigate the use of both gestures at once, so that users have the option to choose their preferred gesture for different controls. For example, users could use AirPush for discrete activations and Pinch for continuous actions like slider control.

7.4 Practical Use of Proxemic Cursors

A potential limitation of a proxemic cursor is that it allows users to start gesturing in a poor sensing position, with further movements (i.e., after taking control of the slider) resulting in the hand moving even further away from the 'sweet spot' [1, 13] where sensing is most reliable. This is not as much an issue with the buttons but will affect sliders where relative displacement is part of the interaction. As shown in Figure 9 and Figure 14, users generally still moved the cursor towards the slider handle before taking control and towards the button bounding box. Since our implementation used an absolute mapping of hand-to-cursor position, this had the effect of drawing the hand into a good position for sensing. Our users did not 'clutch' to reposition the cursor mid-task, though we used a wide-FOV input sensor (Ultraleap Stereo IR 170). However, in situations where sensor range or FOV are limited, or there are concerns about input sensing reliability, it may be necessary to artificially constrain the size of the proxemic cursor activation zones to encourage users to remain within the sensing sweet spot. Similarly, if handle displacement is likely to cause sensing issues, limiting slider width and positioning more centrally on the screen will limit the distance the hand moves from the sweet spot.

Our findings present clear evidence of the benefits of proxemic cursor widgets for touchless slider input, and we recommend touchless interface designers incorporate these into their interaction vocabulary. We took a rudimentary approach to visual feedback design for this formative evaluation of the concept, using the same visual feedback for proxemic and non-proxemic cursors. However, more sophisticated designs could further enhance the efficacy of proxemic cursor interactions and this is a compelling area for future research. For example, adaptive cursor appearance (inspired by, e.g., BubbleCursor [16]) or additional feedforward [6, 39] could improve usability over our basic designs. Adding other modalities like gaze [4, 25] could also bring potential benefits, e.g., reducing the amount of necessary hand movement.

8 CONCLUSION

In this paper, we explored touchless proxemic cursor interactions, supporting more efficient and comfortable mid-air touchless interactions. Across two studies, we explored the utility of targeting widgets based on their proximity to a hand-projected cursor.

Our experimental results characterised the effect of proxemic cursor interactions during selection tasks, showing how users utilised proxemic targeting to improve input performance. Proxemic cursors led to faster task completion, largely influenced by the reduced time needed to acquire control of slider widgets and button targeting, and reduced time to move to another widget. Users still tended to move the cursor towards widgets as a means of avoiding targeting ambiguity, but used proxemics for faster and more relaxed targeting, especially when it reduced arm movement or when widgets had reduced surface area. Proxemic cursor interactions also reduced task workload and were most preferred by users, especially when using Pinch as an activation gesture. Pinch is emerging as a promising alternative to AirPush for mode switching, but our results show that proxemic cursor behaviour can also improve the usability of AirPush, which is still the predominant gesture in many touchless gesture interfaces (e.g., Ultraleap TouchFree [34]).

In summary, this work shows the potential benefits of proxemic cursor widgets, especially for slider input. As touchless technology continues to grow and reach new application domains, addressing the usability of fundamental input operations like sliding is important. This will help to close the performance gap between touch and touchless, and can further open up the technology for more sophisticated and expressive mid-air interactions.

REFERENCES

[1] Florian Alt, Andreas Bulling, Gino Gravanis, and Daniel Buschek. 2015. GravitySpot: Guiding Users in Front of Public Displays Using On-Screen Visual Cues. In

Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 47–56. https://doi.org/10.1145/2807442.2807490

[2] Myroslav Bachynskyi and Jörg Müller. 2020. Dynamics of Aimed Mid-Air Movements. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376194

[3] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300331

[4] Ishan Chatterjee, Robert Xiao, and Chris Harrison. 2015. Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (Seattle, Washington, USA) *(ICMI '15)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2818346.2820752

[5] Christopher Clarke and Hans Gellersen. 2017. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 179–192. https://doi.org/10.1145/3126594.3126626

[6] Sven Coppers, Kris Luyten, Davy Vanacken, David Navarre, Philippe Palanque, and Christine Gris. 2019. Fortunettes: Feedforward about the Future State of GUI Widgets. *Proc. ACM Hum.-Comput. Interact.* 3, EICS, Article 20 (jun 2019), 20 pages. https://doi.org/10.1145/3331162

[7] Patricia Ivette Cornelio Martinez, Silvana De Pirro, Chi Thanh Vi, and Sriram Subramanian. 2017. Agency in Mid-air Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2426–2439. https://doi.org/10.1145/3025453.3025457

[8] Sean Cronin, Euan Freeman, and Gavin Doherty. 2022. Investigating Clutching Interactions for Touchless Medical Imaging Systems. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3517512

[9] Ashley Dover, G. Michael Poor, Darren Guinness, and Alvin Jude. 2016. Improving Gestural Interaction With Augmented Cursors. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (Tokyo, Japan) *(SUI '16)*. Association for Computing Machinery, New York, NY, USA, 135–138. https://doi.org/10.1145/2983310.2985765

[10] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. 2021. An Aligned Rank Transform Procedure for Multifactor Contrast Tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 754–768. https://doi.org/10.1145/3472749.3474784

[11] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2014. Tactile Feedback for Above-Device Gesture Interfaces: Adding Touch to Touchless Interactions. In *Proceedings of the International Conference on Multimodal Interaction - ICMI '14*. ACM, 419–426. https://doi.org/10.1145/2663204.2663280

[12] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems - CHI '16*. Association for Computing Machinery, 2319–2331. https://doi.org/10.1145/2858036.2858308

[13] Euan Freeman, Dong-Bach Vo, and Stephen Brewster. 2019. HaptiGlow: Helping Users Position their Hands for Better Mid-Air Gestures and Ultrasound Haptic Feedback. In *2019 IEEE World Haptics Conference (WHC)*. IEEE, 289–294. https://doi.org/10.1109/WHC.2019.8816092

[14] Euan Freeman, Julie Williamson, Sriram Subramanian, and Stephen Brewster. 2018. Point-and-Shake: Selecting from Levitating Object Displays. In *Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Paper 18. https://doi.org/10.1145/3173574.3173592

[15] Charles P. Gerba, Adam L. Wuollet, Peter Raisanen, and Gerardo U. Lopez. 2016. Bacterial contamination of computer touch screens. *American Journal of Infection Control* 44, 3 (2016), 358–360. https://doi.org/10.1016/j.ajic.2015.10.013

[16] Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 281–290. https://doi.org/10.1145/1054972.1055012

[17] Darren Guinness, Alvin Jude, G. Michael Poor, and Ashley Dover. 2015. Models for Rested Touchless Gestural Interaction. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (Los Angeles, California, USA) *(SUI '15)*. Association for Computing Machinery, New York, NY, USA, 34–43. https://doi.org/10.1145/2788940.2788948

[18] Aakar Gupta, Thomas Pietrzak, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. 2017. Summon and select: Rapid interaction with interface controls in mid-air. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces - ISS '17*. Association for Computing Machinery, 52–61. https://doi.org/10.1145/3132272.3134120

[19] Carl Gutwin. 2002. Improving Focus Targeting in Interactive Fisheye Views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI '02)*. Association for Computing Machinery, New York, NY, USA, 267–274. https://doi.org/10.1145/503376.503424

[20] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[21] Morten Hertzum and Kasper Hornbæk. 2007. Input techniques that dynamically change their cursor activation area: A comparison of bubble and cell cursors. *International Journal of Human-Computer Studies* 65, 10 (2007), 833–851. https://doi.org/10.1016/j.ijhcs.2007.05.001

[22] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1063–1072. https://doi.org/10.1145/2556288.2557130

[23] Intel. 2021. Intel RealSense Touchless Control Software. https://www.intelrealsense.com/introduction-to-intel-realsense-touchless-control-software Last Accessed 13/09/2022.

[24] Mikkel R. Jakobsen, Yvonne Jansen, Sebastian Boring, and Kasper Hornbæk. 2015. Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, Cham, 455–473.

[25] Mathias N. Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Grønbæk, and Hans Gellersen. 2022. Gaze-Hand Alignment: Combining Eye Gaze and Mid-Air Pointing for Interacting with Menus in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* 6, ETRA, Article 145 (may 2022), 18 pages. https://doi.org/10.1145/3530886

[26] Ville Mäkelä, Jonas Winter, Jasmin Schwab, Michael Koch, and Florian Alt. 2022. Pandemic Displays: Considering Hygiene on Public Touchscreens in the Post-Pandemic Era. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 12 pages. https://doi.org/10.1145/3491102.3501937

[27] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. 2010. Clutch-Free Panning and Integrated Pan-Zoom Control on Touch-Sensitive Surfaces: The Cyclostar Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2615–2624. https://doi.org/10.1145/1753326.1753724

[28] Jörg Müller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. 2012. Looking Glass: A Field Study on Noticing Interactivity of a Shop Window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, 297–306. https://doi.org/10.1145/2207676.2207718

[29] Jennifer Sarah Pearson, Gavin Bailey, Simon Robinson, Matt Jones, Tom Owen, Chi Zhang, Thomas Reitmaier, Cameron Steer, Anna Carter, Deepak Sahoo, and Dani Kalarikalayil Raju. 2022. Can't Touch This: Rethinking Public Technology in a COVID-19 Era. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. https://doi.org/10.1145/3491102.3501980

[30] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-Touch: Combining Gaze with Multi-Touch for Interaction on the Same Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 509–518. https://doi.org/10.1145/2642918.2647397

[31] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental Analysis of Barehand Mid-Air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300426

[32] Ultraleap Ltd. 2021. LEGO: Touchless Interactive Play Experience. https://www.ultraleap.com/company/news/case-study/lego-interactive-experience/ Last Accessed 13/09/2022.

[33] Ultraleap Ltd. 2021. Touch free exhibit arrives at the Aquarium of the Pacific. https://www.ultraleap.com/company/news/press-release/touchfree-aquarium-pacific-exhibit/ Last Accessed 13/09/2022.

[34] Ultraleap Ltd. 2021. TouchFree For Kiosks and Digital Signage. https://www.ultraleap.com/enterprise/touchless-experiences/touchfree-solution Last Accessed 11/07/2022.

[35] Ultraleap Ltd. 2021. Touchless Interactive Kiosks: The Air Push Gesture. https://www.ultraleap.com/company/news/blog/touchless-interactions/ Last Accessed 13/09/2022.

[36] Ultraleap Ltd. 2022. PepsiCo innovation: Touch less, smile more! https://www.ultraleap.com/company/news/case-study/pepsico-touchless-ordering/ Last Accessed 13/09/2022.

[37] Ultraleap Ltd. 2022. UST Integrates Ultraleap Tech To Power Faster Checkout Experiences. https://www.ultraleap.com/company/news/press-release/ust-checkout-experience/ Last Accessed 13/09/2022.

[38] Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3, Article 22 (apr 2017), 35 pages. https://doi.org/10.1145/3064937

[39] Jo Vermeulen, Kris Luyten, Elise Hoven, van den, and Karin Coninx. 2013. Crossing the bridge over Norman's gulf of execution : revealing feedforward's true identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2013 (CHI '13)*. Association for Computing Machinery, 1931–1940. https://doi.org/10.1145/2470654.2466255

[40] Kieran Waugh, Mark McGill, and Euan Freeman. 2022. Push or Pinch? Exploring Slider Control Gestures for Touchless User Interfaces. In *Proceedings of the 2022 Nordic Human-Computer Interaction Conference (NordiCHI '22)*. Association for Computing Machinery. https://doi.org/10.1145/3546155.3546702

[41] Kieran Waugh, Mark McGill, and Euan Freeman. 2023. *User Study Data for "Proxemic Cursor Interactions for Touchless Widget Control"*. https://doi.org/10.5281/zenodo.8286402

[42] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[43] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. 2020. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376876

[44] Futian Zhang, Sachi Mizobuchi, Wei Zhou, and Edward Lank. 2021. Analyzing Midair Object Pointing Mappings for Smart Display Input. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 490 (nov 2021), 18 pages. https://doi.org/10.1145/3488535