A Study of Multimodal Pen + Gaze Interaction Techniques for Shape Point Translation in Extended Reality

Uta Wagner*
University of Konstanz, Germany

Jinwook Kim†
KAIST, Korea

Zhikun Wu‡
KTH Royal Institute of Technology, Sweden

Qiushi Zhou§
Aarhus University, Denmark

Mario Romero¶
Linköping University, Sweden

Alessandro Iop‖
KTH Royal Institute of Technology, Sweden

Tiare Feuchtner**
University of Konstanz, Germany.

Ken Pfeuffer††
Aarhus University, Denmark

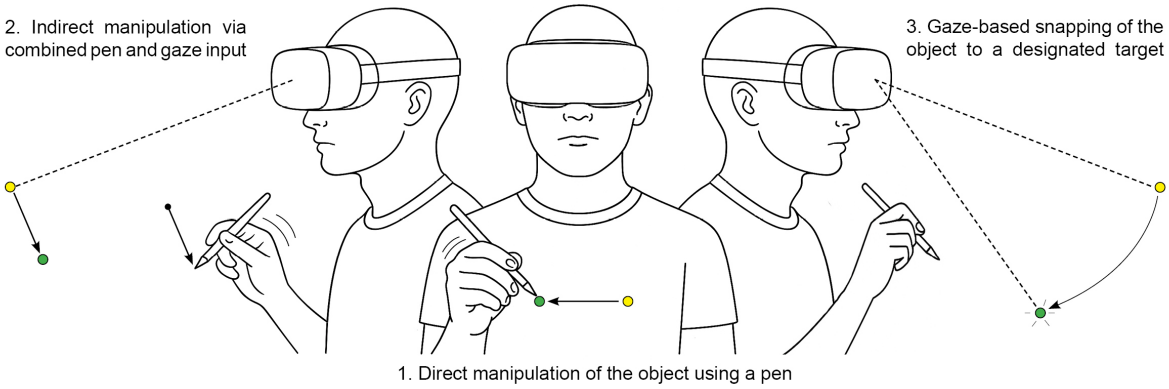


Figure 1: Overview of three manipulation techniques in virtual reality utilizing pen and gaze input: (1) DirectPen, (2) Gaze + Pen, and (3) GazeSnap.

Abstract

Eye-tracking offers new ways to augment our interaction possibilities in extended reality. This paper investigates how gaze can assist pen users in translating shape points within graphical models. By leveraging gaze, we can support the usual design activities with an option where objects can be selected and repositioned through eye movements, with the pen serving as a confirmation tool. This can reduce manual effort and enhance efficiency and ergonomics. To evaluate its effectiveness, we compare four interaction techniques: two pen-based baselines (direct and ray-based) and two gaze-supported methods (gaze for selection and/or object dragging), using a probability based selection scheme. In a user study, 16 participants carried out a shape point translation task and their performance, effort, and user experience were measured. The results highlight the performance trade-offs of each technique—while the gaze-based dragging method introduced marginally more errors, it significantly reduced task time. Our findings offer comparative insights into the strength and limitations of gaze-and pen-based interaction methods, supporting the design of future multimodal 3D design tools.

*e-mail: uta.wagner@uni-konstanz.de
†e-mail: jinwook.kim31@kaist.ac.kr
‡e-mail: zhikun@kth.se
§e-mail: qiushi.zhou@cs.au.dk
¶e-mail: mario.romero@liu.se
‖e-mail: aiop@kth.se
**e-mail: tiare.feuchtner@uni-konstanz.de
††e-mail: ken@cs.au.dk

Index Terms: Input techniques, vr, eye-tracking, gaze interaction.

1 Introduction

In Extended Reality (XR) environments, 3D design and modelling tools (e.g., Gravity Sketch, Tilt Brush, ShapesXR) support people by enhancing creativity, spatial awareness, and prototyping. One of the most suitable input devices in this domain is the stylus. Users can intuitively perform a range of design tasks from writing, drawing strokes, and creating shapes, up to large-scale architectural modelling of complex systems. We explore how interactions with the stylus can be advanced through multimodal gaze input.

Gaze interaction is established for XR headsets (e.g., based on the AndroidXR, Apple Vision Pro, or Meta Quest Pro), often combined with pinch gestures to interact with distant objects (”Gaze + Pinch” [29, 33, 42]). A key concept underpinning this interaction model is the duality of direct and indirect input modes in the XR UI—users can either perform direct gestures within reachable space or employ Gaze + Pinch otherwise. Similarly, pens could be augmented with multimodal gaze input capabilities to support design tasks [30]. However, given the multifaceted nature of 3D design work, it is unclear how gaze might effectively support such tasks, thus highlighting the need for more research grounded in design-specific use cases [40].

As a first step, we focus on a particular, but common task in working with graphical shapes: *shape point translation*. Shapes are composed of nodes and edges that form a relational graph, which users manipulate by selecting and translating nodes to reshape the object [22]. By default, the pen naturally serves as a direct manipulation tool, for instance for creating new nodes, editing them, and establish lines between nodes. Typically, shape point translation is done by absolute pointing at a node [35] or grabbing it directly with the pen and moving it to a new location in space. However, when working with large shapes that extend beyond arm's reach — or in

tasks that require frequent, repetitive node selections — this process can become physically tiring. In these scenarios, it can be helpful to use gaze alongside the pen to perform the tasks more efficiently.

In this paper, we investigate how gaze-based interactions can complement the pen in shape point translation for 3D design tasks. Specifically, we investigate the following gaze-based interaction techniques. The first, Gaze + Pen, uses gaze for selecting a target node, while manipulation is performed with the pen (Figure 1-left). This follows the widely used Gaze + Pinch model [33, 42, 29], adapted here for pen input. We compare this multimodal technique to two common baselines: direct pen input, where users physically reach each node, and pen-based raypointing, where a ray projects from the pen tip to select nodes at a distance.

In addition, we investigate GazeSnap, a novel technique that involves gaze input more extensively than Gaze + Pen throughout the interaction (Figure 1-right). While it builds on prior work by Wagner et al. on 3D object movement [41], our focus shifts to shape point translation, where users interact with predefined nodes on existing objects rather than positioning entire objects in space. As this task emphasizes discrete point selection rather than continuous 3D targeting, it aligns well with object-based interaction models [18, 17]. Prior gaze interaction research has shown that snapping mechanisms — where gaze input is anchored to nearby selectable elements — can improve both accuracy and usability by better reflecting how we naturally attend to objects and by reducing the effects of jittery gaze movements [33]. For these reasons, we explore how gaze-based snapping compares to other techniques in supporting precise and repetitive shape editing tasks.

We present a user study that compares the four interaction techniques for shape point translation in 3D design. We designed a node-connection task, where the user first selects an origin node and then moves this to a destination point. The techniques for comparison include (1) DirectPen, (2) Pen-ray, (3) Gaze + Pen, and (4) GazeSnap. The integration of GazeSnap into a node-based translation task introduces new trade-offs worth comparing. To isolate input-related effects, we structured the study around a single-point translation task, evaluating user performance and usability of the techniques across 2 target sizes and 3 reachable distances. While the advantages of gaze include interaction over distance, we were also interested in how it compares to DirectPen input, and thus a consistent set of targets within arms reach was used. While situated within the broader domain of 3D shape editing, our study specifically focuses on a single-point translation task. This controlled scope provides a foundation for future studies that address more complex shape manipulation operations such as inserting or removing nodes or modifying edge types.

Our Research Questions are **(RQ1)**: *What is the effect of integrating gaze in a pen-based shape point translation task?* Secondly, **(RQ2)**: *What is the effect of snapping vs. a typical model of "gaze selects, hands manipulate"?* .

Our findings highlight distinct performance trade-offs across techniques. GazeSnap reduced task time, with users completing the task in 2.5 seconds on average, compared to 3.41 to 3.64 seconds for the baseline techniques. In contrast to prior work [41, 28], this firstly shows a substantial temporal improvement result for eye-hand techniques of about 30% in a manipulation task. As expected, it also led to the lowest perceived physical effort and task load. The performance advantages traded with a minor effect on error rate of 2.6 %, whereas other techniques ranged from 0.5 to 1.2 %. The lower physical load was traded with an increase in eye fatigue, too, supporting earlier results [34]. In contrast, DirectPen input was slower and more physically demanding, but yielded lower error rates. These findings highlight that while GazeSnap offers strong effects for increased speed and reduced effort, it introduces new challenges around precision and sustained visual strain — important considerations for integrating gaze into 3D design workflows.

2 Related Work

We summarise previous work on multimodal pen input, pen input in XR environments, and the combined use of pen and gaze that inspired the design and evaluation of our proposed techniques.

2.1 Multimodal Pen Input

Using the pen in a multimodal fusion has been extensively explored. For instance, prior work has combined pen input with gestures or touch, inspired by real-world manual behavior. Wu et al. [45] showed that multi-hand gestures enhance pen input expressiveness. Hinckley et al. [19] introduced the "pen writes, touch manipulates" principle, using the Non-Dominant Hand (NDH) to support pen use. Yee et al. [46] proposed asymmetric pen-touch input for a more natural experience. Brandl et al. [9] demonstrated that combined input improves speed and accuracy, and introduced NDH-based mode-switching for sketching.

Previous work also explored rendering the pen more expressive for 2D and 3D spaces. For instance, "Drag-and-Pop" enables cross-screen dragging of virtual elements using a physical pen device, while "Drag-and-Pick" further facilitates this action by actively bringing all target icons close to the pen-point for dropping [37, 7]. Most designs of multitouch-pen input were inspired by bi-manual input techniques, such as symmetric two-handed spline manipulation using two mice [26], and throw-and-catch of virtual objects through markerless hand tracking [43]. "Bi-3D" demonstrated bi-manual pen and touch interaction for 3D manipulation on 2D tablets, with the NDH managing RST (Rotate, Scale, Translate) manipulation of the target object, while the pen performed precise selection and manipulation of control points to a 3D structure [32]. The combination of multitouch and pen has been found to support similar interactions that involve the selection, manipulation, and connection of nodes, which form the backbone of most pen-based interaction with digital interfaces other than drawing [23]. In this work, we investigate multimodal pen input in XR for a similar task of control point manipulation and connection [32, 23], while featuring gaze-based interactions inspired by previous research, which proactively connects the dragged object and its destination target for drag and drop tasks [7, 12, 43].

2.2 Pen Interfaces in XR

Whereas previous work explored mid-air pen input with early desktop Virtual Reality (VR) systems [13], the large virtual spaces afforded by modern XR systems still challenge effective uses of mid-air pen input due to fatigue and limited accuracy [24, 38]. Early work aimed to address this issue by using a physical 2D surface, typically a tablet device. For example, Arora et al. proposed a number of such applications and found that pen input performance significantly improves with a physical drawing surface [2] that can support free-form mid-air pen sketching in XR [1]. Other works explored incorporating multitouch gestures on tablets, to further facilitate pen input in XR [15]. For instance, VRSketchIn investigated a design space of pen and tablet interaction for 3D sketching in VR that combines unconstrained 3D mid-air with constrained 2D surface-based sketching [14].

Previous explorations have shown that even without a physical surface, pen input in 3D XR environments could improve pointing performance. Though pens can mimic VR controllers for pointing [5, 35], natural pen use is typically studied in design contexts, where interaction demands differ from general UI control. Zou et al. investigated the effect of interface types on drawing accuracy and user comfort in XR, and found that the majority of users preferred holding a tool for drawing, simulating the pen-drawing experience in physical environments [49]. In a different study, they found that the asymmetric stylus and gesture inputs, using the NDH with the DH, exhibit favourable usability and facilitate fast and accurate VR sketching [50]. The familiarity of holding a pen in

physical environments may be transferred to XR through the pen-gripping posture that facilitates precision. In an evaluation of pen grip gestures for VR input, Li et al. found that a "tripod" grip (how most people grip pens) at the rear end of the pen outperforms traditional wrist-based input both for direct input using the pen tip and for indirect input using a pen-ray [27]. Similarly, Batmaz et al. found that the "precision grip" (equivalent to tripod grip) significantly improved the accuracy of user performance in VR [6]. Chen et al. found that, compared to using bare hands, VR controllers and pens yield significantly higher precision and user performance in a 3D target tracing task. They suggest that, while a VR pen can benefit precise 3D drawing, coarse–grained tasks (e.g., target selections) can be allocated to hand or another pointing modality, because using the pen may induce higher fatigue [10]. In our present work, we explore different combinations of gaze pointing and pen input that benefit from the distance-reaching capabilities of gaze input to complement the precise pen input.

2.3 Combining Pen and Gaze for Input

Gaze is an important input modality that has been explored to complement pen input since the early tablet-based works. For instance, Pfeuffer et al. proposed Gaze-Shifting that enabled direct-indirect input with pen and touch that is modulated by gaze pointing at different regions on the touchscreen for a shape editing task [30]. They later studied pen + gaze techniques for a compound pan, zoom, and ink task, finding that it leads to comparable performance as using default pen + touch inputs [31]. Gaze + Pinch was proposed in a later work following a similar direct-indirect gaze modulation approach for using pinch gestures to select and manipulate virtual objects in a 3D XR environment, which has since evolved into a standard approach for 3D interaction in XR[1] [33]. Similar to earlier work on gaze and pen [30, 31], Gaze + Pinch benefits from the capability of gaze to quickly access targets across large distances.

While previous research has explored the distance-crossing benefit with pen input for drag-and-drop across physical displays [37, 7], similar use of gaze pointing for drag-and-drop remains less explored. An early work, Eyedraw, explored using gaze for drawing pictures and suggested that gaze drawing can be used for coarse drag-selection tasks on traditional 2D UIs [20]. Inspired by Magic pointing [47], MaRginalia enabled an XR note-taking application in which gaze and pen collaboratively manipulate a cursor across multiple virtual windows [36]. A recent study [41] that directly explored and evaluated the use of gaze for drag-and-drop operations found that *Look&Drop*, a division of labour between gaze and hand each controlling different degrees-of-freedom (DOF) of target movement, yielded lower physical effort and greater user preference than direct gestures [41]. However, the study did not yield benefits in overall task time. Specifically, gaze led to decreased object selection, but increased object dragging time. Similarly, Lystbaek et al.'s work on Hands On, Hands Off [28] showed no clear performance advantage over direct inputs.

In the present work, we investigate how a "gaze-drags, pen confirms" technique, with the difference that our point of departure is the pen in the user's hand, which has different affordances than hand gestures. Further, we include a gaze+pen technique that uses target-snapping, a method that has been shown highly useful in object based interfaces [17, 18], including gaze and pinch input [33].

3 Supporting Gaze-based Pen Interaction

In this paper, we compare the pen-based ray technique and gaze-based pen techniques for mid-air interactions with a reference method for object selection and target placement of varying sizes and distances. Each technique follows four steps: *Indicate, Confirm, Manipulate, Release*. The techniques differ in their modality and input structure, such as pointing with gaze versus pen.

[1] https://support.apple.com/guide/apple-vision-pro

In the **Object Selection phase**, the user points to the object using the pen (DirectPen), pen-ray (Pen-ray), or gaze (Gaze + Pen and GazeSnap). Once the object enters the hover state, it indicates the object is selectable. The user then holds the trigger button, and the object enters the dragging state.

In the **Dragging phase**, once the object is in the dragging mode, its position is controlled by the pen - for DirectPen, Pen-ray and Gaze + Pen- not by gaze, which is only used for pointing. However, with GazeSnap, manipulation is done using gaze, while the pen is only used for confirmation.

Object selection and Target placement can occur in various ways: The first option involves selecting with the pen (DirectPen and Pen-ray), the second with gaze (Gaze + Pen and DirectPen).

When moving and dragging with gaze (GazeSnap), the user looks at the target, enabling full 3D pointing with the gaze. The selection is made with target knowledge, meaning the user needs to keep the target in view. If the user fails to maintain focus on the target with their gaze, the last gaze position is used to determine and place the target.

3.1 Techniques

3.1.1 DirectPen (Figure 2: a-c)

Serves as a baseline for direct interaction with objects in mid-air using a pen. A button press gives the user physical feedback during selection, making the interaction more deliberate and precise. Additionally, the technique supports pen dragging and allows flexible use in various 2D and 3D interaction scenarios. This makes it particularly suitable for tasks such as drawing, writing, or sketching. We use the implementation provided by Meta Quest Pro, which uses the absolute position of the pen tip as the selection point. This approach, which does not use additional pointing enhancements, aligns with how direct pen manipulation is used in 2D and 3D contexts in industry and academia [19, 22, 30, 34].

3.1.2 Pen-ray (Figure 2: d-f)

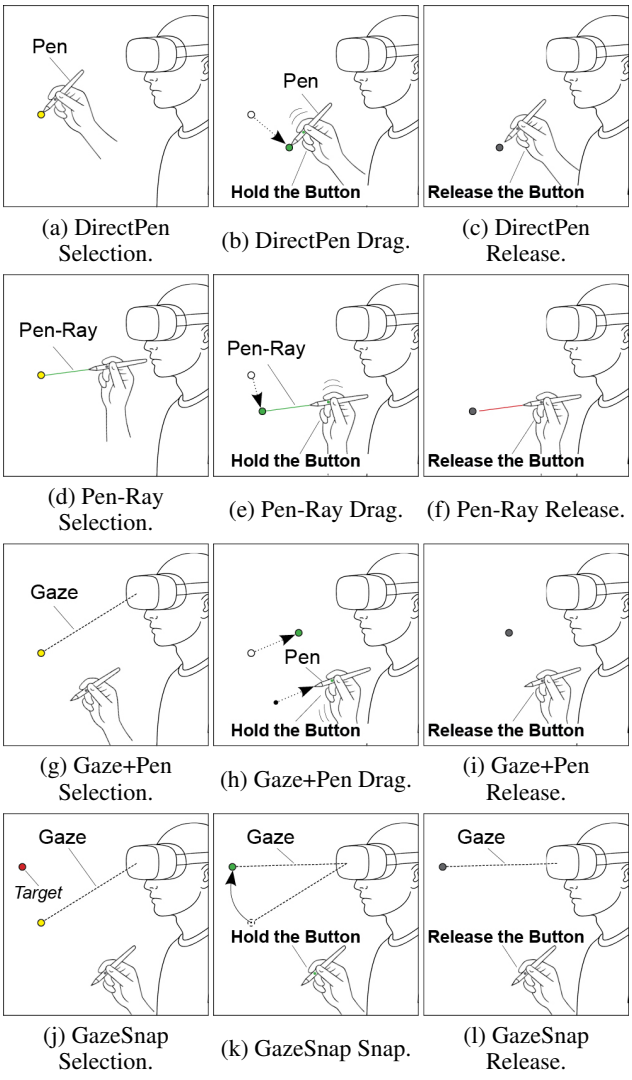
The combination of Pen-ray interaction is ideal for precise pointing and continuous input, similar to the metaphor of a laser pointer. This indirect laser pointer interaction (pen ray + pen button) becomes even more effective when the ray is directed toward the user's line of sight. It allows for rapid pointing and object movement along the Y-axis. The pen button confirms the selection, while the indirect pen ray enables precise pointing and dragging of objects. This approach enhances control and efficiency in 3D environments, aligns with industry usage (e.g., as used in the Logitech VR pens) providing smooth, intuitive interaction without the need for direct physical contact.

3.1.3 Gaze + Pen (Figure 2: g-i)

The Gaze + Pen interaction allows objects to be selected from a distance and manipulated with the pen once the pen button is pressed. The technique works on the principle of *gaze selects, hand with pen manipulates*. Gaze is used for pre-selection, the pen button confirms the selection, and while the button is pressed, the object can be manipulated through indirect pen movements. The combination of gaze-based pen-button feedback provides an intuitive and effortless way to move objects in any direction. This allows objects to be adjusted along all three axes (X, Y, Z) in space without relying on traditional interaction methods.

3.1.4 GazeSnap (Figure 2: j-l)

The GazeSnap technique is a concept that enables object selection and target placement using gaze: The object is selected by the user's eyes, and the selection is confirmed by pressing the pen button. Afterward, the user focuses on the target object. Once the pen button is released, the object is placed at the target location. When using this technique, the user moves or drags the object by gazing at the



(a) DirectPen Selection. (b) DirectPen Drag. (c) DirectPen Release.

(d) Pen-Ray Selection. (e) Pen-Ray Drag. (f) Pen-Ray Release.

(g) Gaze+Pen Selection. (h) Gaze+Pen Drag. (i) Gaze+Pen Release.

(j) GazeSnap Selection. (k) GazeSnap Snap. (l) GazeSnap Release.

Figure 2: Illustration of four interaction techniques: DirectPen (a-c), Pen-Ray (d-f), Gaze+Pen (g-i), and GazeSnap (j-l).

destination. If the gaze is within the target's bounds, visual feedback shows that the entire object can be selected (hover state). As such, different to the other techniques, this technique is based on target knowledge in order to enable the snapping.

In our pilot tests, we first tried a simple way to let the user select an object and then a target with their gaze. But this naive approach led to problems: users often lost their fixations, and jitter made it hard to lock onto the target. As a result, the error rate was high. Simply enlarging target sizes could address the issue, but this approach is hard to generalize, as target layouts often cannot be altered. To solve this, we added the *1€ Filter* and a *Fixation Filter* to smooth the raw gaze data and reduce small fluctuations. Inspired by earlier work [39], we also introduced a *Probability-Based Gaze Validation* mechanism to handle any remaining uncertainties and confirm the user's intent more reliably. These steps improved the stability of our gaze snapping technique, and we discuss the details in the next section.

3.2 Technical details

For the Experimental setup, we used the Meta Quest Pro (106°x95.57° FOV, 1800x1920 pixels per eye), and the software was implemented with the OVR toolkit in Unity3D (2022.3.12f1). Eye-tracking accuracy is reported to be around 1.5-3° [44, 3].

We used several filtering and validation mechanisms to process the gaze data. First, we used a *1€ Filter* to smooth raw gaze input and reduce noise-related fluctuations, based on [48]. The parameters were set to $f_{c_{min}} = 1.5$ and $\beta = 20$ for gaze, and $f_{c_{min}} = 0.9$ and $\beta = 90$ for controller movements. Additionally, we used a *Fixation Filter* to distinguish stable fixations from rapid saccades. The fixation angle was set to 4, and 0.25s was used for fixation time. To address the instability of gaze selection, we also implemented a *Probability-Based Gaze Validation* mechanism that collects per-frame gaze data over a short time window (300*ms*) rather than relying solely on single-frame data, only entering a hover state if more than 50% of the recorded gaze hit within the target object.

For pen input, we used the *Meta Quest Pro Right Controller* with a *stylus tip* and the *grip button* for interaction. The controller's input data, including both position and rotation, was mapped onto a virtual pen, replicating the behavior of the *Logitech XR Pen* to ensure consistency with the pen input system [25]. The GazeSnap algorithm for drag interactions was customized to handle gaze-based dragging and calculate drag movements based on gaze position. This algorithm also compensates for head movements to maintain accurate object dragging. We adopted the *Control-Display (CD) ratio* to regulate drag acceleration. This method applies a quadratic function, which amplifies larger movements more than smaller ones, based on the work of Wagner et al.[41].

4 User Study

We conducted an empirical user study to evaluate the usability of the proposed interaction techniques for shape point translation in 3D space. The study assesses selection and manipulation tasks with Gaze+Pen input. Our baseline is the direct use of the pen (DirectPen). The study task involves repositioning an object to connect two shape points, followingdiagonal dragging directions, inspired by previous work [4, 41]. This design reflects a constrained yet relevant subset of shape construction task in design contexts. Our research questions include

(RQ1): *What is the effect of integrating gaze in a pen-based shape point translation task?* Prior work has shown that gaze can complement manual input, but its benefits in shape construction tasks remain underexplored. Here we compare two gaze-based techniques to two pen-based techniques to understand performance and usability differences.

Secondly, **(RQ2)**: *What is the effect of snapping vs. a typical model of "gaze to select, hands to manipulate"?* Different gaze integration possibilities can lead to different interaction dynamics. Here we compare GazeSnap, which uses automatic snapping to Gaze + Pen, which uses gaze for preselection without snapping, and is based on Gaze+Pinch [33]. Moreover, the two approaches may offer different trade-offs in speed and control.

To achieve a balanced experimental design, we counterbalanced the order of the four techniques across participants. Within each block, both Distances and Target Sizes were randomly varied. In total, the study yielded 16 *Participants* × 4 *Techniques* × 3*Distances* × 2 *Target Sizes* × 8 *repetitions* = 3072 data points.

4.1 Task

The study task required users to select an object and move it to a target. The object and target were randomly placed in diagonally opposite corners [41]. Additionally, we decided to test the objects and targets in two sizes (4° [0.0395 m] and 6° [0.0595 m]) and at three different distances: *near* (15 cm), *mid-range* (35 cm), and *hard of reach* (55 cm), with conditions randomized within each task block. This ensured that users had to extend their arms or engage in more body movement for farther targets than closer ones. Each user completed a total of eight repetitions [4, 41].

Using the assigned technique, the user was instructed to place the object at the target position as quickly and accurately as possible. If the target was not reached within 30 seconds, it was counted as a time-out and considered an error. An error was also recorded if the user performed a double button press or lost the object during the dragging process. A double click was treated as an error when it caused unintended behavior—such as selecting and immediately deselecting the target—leading to trial cancellation and repetition.

Additionally, both the object and the target were embedded within geometric shapes, giving users the impression of completing a shape. The shapes used were rectangles and triangles, positioned within a cubic volume slightly below eye level. This volume was positioned with a vertical offset of 8.8 cm below the user's eye level, not as a distance from the user's body but as a placement to ensure a consistent and comfortable downward gaze angle during the task. This design served not only to contextualize the task within a point-dragging in a line-drawing context but also to enhance depth perception and spatial understanding. The VR environment itself further reinforced depth perception. Once an object was moved to the target, a new shape appeared.

4.1.1 Visual Feedback (Figure 3)

Users were supported by visual feedback employing a traffic light color metaphor to communicate interaction states, divided in distinct modes: Indicate, Confirm, Manipulate, Release (Figure 3).

Indicate: A translucent red sphere with a red dot at its center is displayed to signal the location of the interactive element.

Confirm - Hover and Selection: Once the user targets the red sphere through gaze fixation (for gaze-based techniques) or by positioning the pen or pen ray over the object, the system enters hover mode. The sphere then changes color to yellow, indicating that it is in a selectable state. To finalize the selection, the user presses the trigger button on the pen device. This explicit action confirms the user's intent and transitions the system into the manipulation phase.

Manipulation-Dragging mode: Upon confirmation, dragging mode is initiated. The object is now movable, and the target sphere (the destination area for object placement) is highlighted using the same translucent red with a central red dot design. This consistent visual language helps the user identify where the object should be moved. As the user guides the object toward the target area, and once it reaches the correct proximity, the target sphere changes its color to yellow, signaling readiness for placement.

Release-Dropping mode: The user completes the placement by releasing the trigger button. If the object is correctly positioned, the system provides immediate positive feedback by changing the visual cue to green, confirming successful task completion. In contrast, if the object is incorrectly placed, the feedback is given in red, indicating an error and prompting the user to retry the task.

4.2 Participants

A total of 16 individuals participated in the study, including 6 women, 2 non-binary individuals, and 8 men, aged between 22 and 35 years (M = 26.63, SD = 3.50). Participants had diverse backgrounds and levels of technical expertise. They self-reported their familiarity with AR/VR/XR on a 5-point Likert scale (1 = little experience, 5 = expert), yielding an average rating of (M = 3.31, SD = 1.40). Experience with eye-gaze interaction was rated at (M = 3.00, SD = 1.51), while controller usage received an average rating of (M = 1.69, SD = 1.15). Regarding handedness, 14 participants identified as right-handed and 2 as left-handed. Additionally, 6 participants wore glasses and 2 used contact lenses during the study.

4.3 Procedure

The study was conducted in a large room where participants sat on a fixed chair. At the beginning, they signed the consent form and completed a demographic questionnaire. They then received an introduction explaining what to expect over the next 45 minutes. After the introduction, the first technique was explained. Before each new technique, a structured process was followed: first, the technique was explained, then — if it was a gaze-based technique — the eye-tracking calibration was performed. Afterward, an initial training session was conducted under random conditions, consisting of eight test trials. Once all explanations and settings were completed, the main study began. Participants were instructed to perform the tasks as quickly as possible. After each study session, they were asked to remove the headset, complete a questionnaire about the technique, and take a short break of 2–3 minutes. This procedure was repeated for all four techniques. After all techniques had been tested, participants were asked to complete a final ranking questionnaire, in which they ranked the four techniques from most to least preferred. Participants were encouraged to rank based on their overall subjective experience.

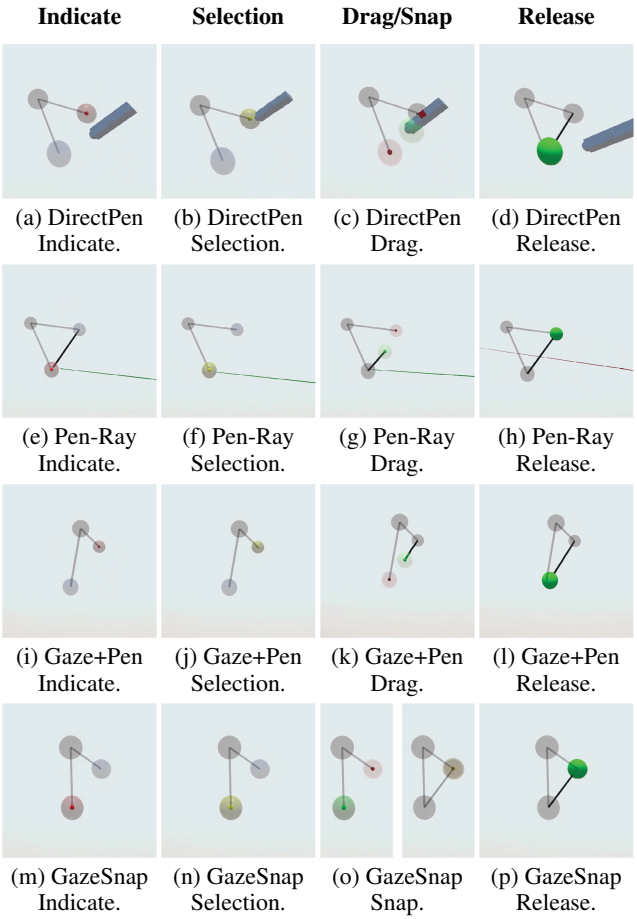
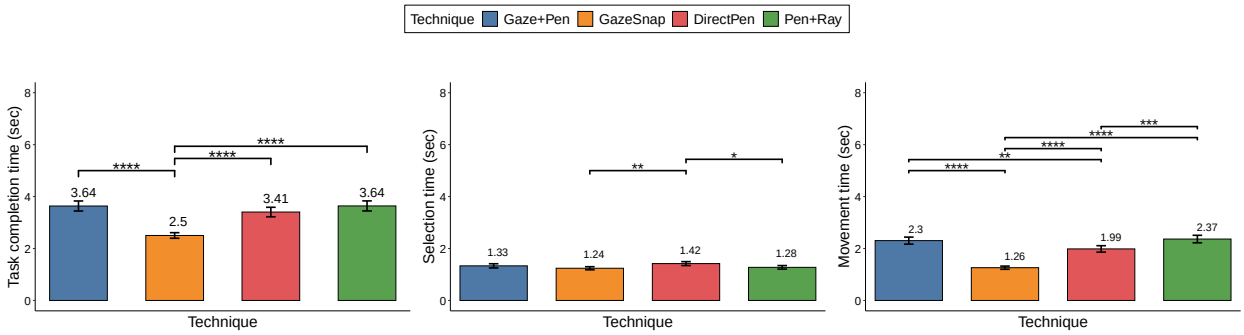


Figure 3: Illustrations of interaction techniques used in the user study: (a-d) DirectPen, (e-f) Pen-ray, (i-l) Gaze + Pen, and (m-p) GazeSnap- with visual feedback shown for the phases of indicate, confirm, manipulate, and release.

4.4 Evaluation Metrics

- **Task Completion Time (TCT)**: Time elapsed between the object appearance and completing the action (button release).
- **Selection Time (ST)**: The time interval from the appearance of the object to the selection action initiated by the pen button press.
- **Movement Time (MT)**: The time elapsed between the completion of the selection action and the release of the button.



(a) TCT. (b) ST. (c) MT.

Figure 4: Time-based performance metrics across four interaction techniques. The four bar charts depict (a) Task Completion Time, (b) Selection Time, (c) Movement Time. Error bars represent 95% confidence intervals.

- **Error Rate (ER)**: A trial is an error either at a timeout or the object exceeds the target radius at the moment of release.
- **Preference and Subjective Task Load**: The NASA-TLX (Task Load Index) questionnaire [11] collects responses, with extra questions on eye and hand fatigue. Participants were asked to submit preference rankings at the study's conclusion, too.

5 Results

We assessed all continuous variables for normality. Where significant deviations were detected, Box-Cox transformations [8] were applied to approximate Gaussian distributions and ensure the validity of subsequent parametric tests. For the subsequent analysis of performance metrics (Sections 5.3 - 5.4), no trials were excluded due to timeouts, defined as failure to complete a selection and target placement task within 30 seconds. Outlier removal was based on Task Completion Time (TCT), where 44 trials (11.458%) exceeded the threshold of the $Mean + 3 \times SD$. A three-way repeated measures ANOVA (Technique × Distance × Target Size) was employed to evaluate the quantitative measures and report generalized eta squared as an appropriate measure of effect size. In cases where the assumption of sphericity was violated, Greenhouse-Geisser adjustments were applied. Post-hoc pairwise comparisons were conducted using estimated marginal means with Bonferroni corrections to control for Type I error. The non-parametric Friedman test was used for Likert-scale data (section 5.5), followed by pairwise Conover post-hoc tests with Bonferroni adjustments. We present statistically significant results concerning the factor *Technique*. Statistical significance is annotated in all graphs using the following notation: * ($p < .05$), ** ($p < .01$), *** ($p < .001$), and **** ($p < .0001$).

5.1 Task Completion Time (Figure 4, Figure 5)

Regarding ***TCT*** ($F^{3.00}_{45.00} = 43.105$, $p < .0001$, $\eta^2_G = 0.34$), we found users were faster in completing the task with GazeSnap than with DirectPen, Gaze + Pen, and Pen-ray ($p < .0001$).

For both factors **Distance** ($F^{1.34}_{20.07} = 278.71$, $p < .0001$, $\eta^2_G = 0.271$) and **Target Size** (F^{1}_{15}=52.398, $p < .0001$, $\eta^2_G = 0.03$) were significant main effects found, which means that users completed the task faster with short Distance of 15*cm* (2.78s, $p < .0001$) than with 35*cm* (3.32s) and 55*cm* (3.79s), and also faster with 35*cm* than with 55*cm* ($p = .0001$). Users were also faster with objects of Target Size of 6° (3.16s, $p = .0098$,) than with 4° (3.43s). Significant interaction effects were found for **Technique** × **Distance** (F^{6}_{90}=2.876,$p = .013$, $\eta^2_G = 0.008$)).

For all three **Distances** (15*cm*, 35*cm*, 55*cm*) and for both **Target Sizes** (4°, 6°), users were significantly faster in completing the task with GazeSnap ($p < .0001$) compared to Gaze + Pen, Pen-ray and DirectPen. There was no significant correlation between error rate and task completion time across techniques and distances ($r = -0.08$), indicating that faster performance did not come at the cost of increased errors.

5.2 Selection Time (Figure 4, Figure 5)

Regarding ***ST*** ($F^{3.00}_{45.00} = 4.92$, $p = .005$, $\eta^2_G = 0.055$), we found users were faster in selecting objects with GazeSnap ($p < .0008$) and Pen-ray ($p < .0030$) than with DirectPen.

For both factors **Distance** ($F^{2}_{30} = 316.351$, $p < .0001$, $\eta^2_G = 0.343$) and **Target Size** (F^{1}_{15}=57.301, $p < .0001$, $\eta^2_G = 0.073$), users selected objects faster with short Distance of 15*cm* (1.1s, $p < .0001$) than with 35*cm* (1.3s) and 55*cm* (1.6s), and also faster with 35*cm* than with 55*cm* ($p < .0001$). Users were also faster with objects of Target Size 6° (1.2s, $p < .0001$) than with 4° (1.4s).

For **Distance** of 55*cm*, users were significantly faster with GazeSnap($p = .0002$) for selecting objects compared to DirectPen.

5.3 Movement Time (Figure 4, Figure 5)

Regarding ***MT*** ($F^{1.95}_{29.31} = 62.41$, $p < .0001$, $\eta^2_G = 0.507$) users were faster in moving the object with GazeSnap than with all the other techniques ($p < .0001$). Also DirectPen was faster compared to Gaze + Pen ($p = .0002$) and Pen-ray ($p = .00003$).

For both factors **Distance** ($F^{2}_{30} = 199.378$, $p < .0001$, $\eta^2_G = 0.187$) and factor **Target Size** (F^{1}_{15}=26.701, $p = .0001$, $\eta^2_G = 0.012$), users moved objects faster with short Distance of 15*cm* (1.69s), than with 35*cm* (2s, $p = .00005$) and 55*cm* (2.24s, $p < .0001$)).

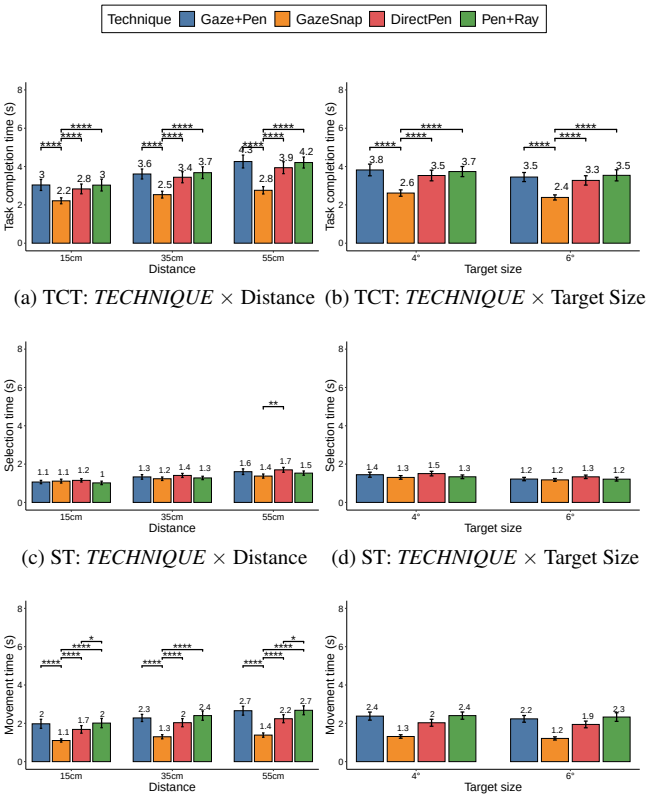
For all three **Distances** (15*cm*, 35*cm*, 55*cm*), users were significantly faster with GazeSnap ($p < .0001$) for moving objects compared to Gaze + Pen, Pen-ray and DirectPen. However, just for 15*cm* ($p = .0049$) and 55*cm* DirectPen was faster compared to ($p = .0079$).

5.4 Error Rate (Figure 7)

Regarding ***ER*** ($F^{1.7}_{25.49} = 4.57$, $p = .025$, $\eta^2_G = 0.049$), users exhibited significantly lower error rate when using DirectPen (0.5%, ($p = .0001$)), Gaze + Pen (0.9%, ($p = .0018$)) and Pen-ray (1.2%, ($p = .0046$)) compared to GazeSnap (2.6%). This effect was particularly pronounced for smaller targets (4°), where DirectPen ($p < .0001$), Gaze + Pen ($p < .0001$), and Pen-ray ($p = .00017$)resulted in significantly lower error rates than GazeSnap.

For both factors **Distance** ($F^{2.0}_{30.00} = 3.47$, $p = .044$, $\eta^2_G = 0.015$) and **Target Size** ($F^{1.0}_{15.00} = 16.68$, $p = .0009$, $\eta^2_G = 0.020$), users made fewer errors with objects of Target Size of 6° (1.8%, $p = .0105$) than with 4° (0.4%).

Significant interaction effects were found for **Technique**×**TargetSize** ($F^{3.0}_{45.00} = 8.65$, $p = .0001$, $\eta^2_G = 0.055$).



(a) TCT: *TECHNIQUE* × Distance (b) TCT: *TECHNIQUE* × Target Size

(c) ST: *TECHNIQUE* × Distance (d) ST: *TECHNIQUE* × Target Size

(e) MT: *TECHNIQUE* × Distance (f) MT: *TECHNIQUE* × Target Size

Figure 5: Mean interaction effects for the four techniques across two factors - *TECHNIQUE* x Distance (left column) and *TECHNIQUE* x Target Size (right column). The rows correspond to (a,b) Task Completion Time, (c,d) Selection Time, and (e,f) Movement Time. Error bars represent 95% confidence intervals; significant differences based on post-hoc analyses are indicated.

For **Distance** (55*cm*), users were significantly more error-prone with GazeSnap ($p < .05$, $p = .00035$) compared to Gaze + Pen; similarly, for **Target Size** (4°), error rates with GazeSnap ($p < .0001$) were significantly higher than with Gaze + Pen ($p < .0001$), Pen-ray ($p = .00017$) and DirectPen ($p < .0001$).

.

5.5 Task Load and Preferences (Figure 6 - Figure 7)

GazeSnap was the most preferred technique, selected by $n = 9$ participants (56.25%), followed by DirectPen with $n = 3$ (18.75%). Both Gaze + Pen and Pen-ray were selected by $n = 2$ participants (12.50%), placing them at the lower end.

A statistical analysis of fatigue ratings revealed significant differences across techniques for both factors **Arm/Hand Fatigue** ($\chi^2(3) = 30.70$, p<.0001, $W = 0.639$) and **Eye fatigue** ($\chi^2(3) = 25.55$, $p < .0001$, $W = 0.532$).

Specifically, Gaze + Pen was rated as less fatiguing for the hand/arm than DirectPen ($p = .00373$), while GazeSnap was perceived as the least fatiguing technique overall, showing a highly significant advantage over all others ($p < .0001$). In contrast, GazeSnap was rated more fatiguing for the eyes than both Pen-ray ($p = .0003$) and DirectPen ($p = .0003$).

The overall **Task load**, as reflected in TLX total scores, also showed significant differences between techniques ($\chi^2(3) = 8.708$, $p < .0001$, $W = 0.181$). GazeSnap ($Mdn = 1.8$) was rated as less demanding compared to Pen-ray ($Mdn = 2.8$, $p < .01$) and DirectPen ($Mdn = 3.2$, $p < .05$).

Perceived **Physical demand**, as measured by NASA TLX scores, differed significantly across techniques ($\chi^2(3) = 25.90$, $p < .0001$, $W = 0.540$). GazeSnap, was rated as less physically demanding ($Mdn = 1$, $p < .0001$) compared to all other techniques: Gaze + Pen ($Mdn = 3$), Pen-ray ($Mdn = 3$), DirectPen ($Mdn = 5$). Additionally, Gaze + Pen was perceived as less physically demanding than DirectPen ($p = .0008$).

A similar result was observed for perceived **Effort**, with a significant overall effect ($\chi^2(3) = 8.548$, $p < .0001$, $W = 0.178$). GazeSnap ($Mdn = 2$) was rated as requiring less effort than DirectPen ($Mdn = 4$, $p = .0056$).

5.6 User Feedback (Figure 6)

The qualitative feedback presented here was collected via open comment fields to supplement the quantitative data and was used exploratively; no formal coding or thematic analysis was performed on these responses. The following quotes serve to illustrate users' subjective experiences and complement the main findings from the NASA-TLX questionnaire and performance metrics.Three participants ranked DirectPen as their favorite technique, while seven considered it their least preferred. Users appreciated its precision and natural interaction style; P12 emphasized it was the *'most accurate and easiest'* technique. The main criticism was the excessive physical effort required, particularly when targeting distant objects. Users reported fatigue and discomfort, with P5 finding it as *'exhausting for the right arm'* and P16 noting it was *'tiring in the long run'*. To cope with these challenges, users developed individual strategies. P15 remarked, *'I notice I will move toward the graph more in this experiment'*, while P13 suggested that *'eye tracking and CD gain significantly will improve these issues'*.

With **Pen-ray** technique, users expressed mixed opinions, with two participants ranking it as their most preferred technique, while five rated it as their least favorite. P1 stated that its performance was *'not good enough'*. P6 and P7 criticized the technique for being unstable when selecting small or distant objects. A common concern was the physical effort involved, especially during interactions at greater distances. P6 described the ray as becoming increasingly unstable the farther the target was, and P10 struggled with controlling the spacing between the ray and the object. Yet, users appreciated the visual feedback provided by the ray and the reduced need for large arm movement compared to DirectPen.

With **Gaze + Pen**, only one participant ranked it as their least preferred technique, while two selected it as their favorite. P13 emphasized liking the balance between gaze and hand input: *'I was still able to move the point manually, while using gaze for initial selection'*. P4 described it as giving a strong sense of control. However, physical strain was experienced, and sometimes, the coordination between gaze and pen was found to be slightly complex. P5 found it *'exhausting to use'*, while P1 noted that involving both eyes and hand made it feel a bit complex. P8 found it easier than Pen-ray since pen movement was only needed for dragging, not selection.

GazeSnap was ranked as the most preferred technique by nine participants. Users appreciated the minimal physical effort required, as gaze was used for both selection and manipulation. P1 described it as the *'most effective and comfortable'* technique, and P6 noted that it felt *'so effortless and quick'*. Participants frequently mentioned reduced hand movement and fast interactions. Some users pointed out limitations. They experienced eye strain or discomfort due to the prolonged use. Furthermore, they have felt the experience lacked a sense of control.

6 Discussion

We studied the effect of integrating gaze control in a pen-based shape point translation task, comparing two gaze-based techniques

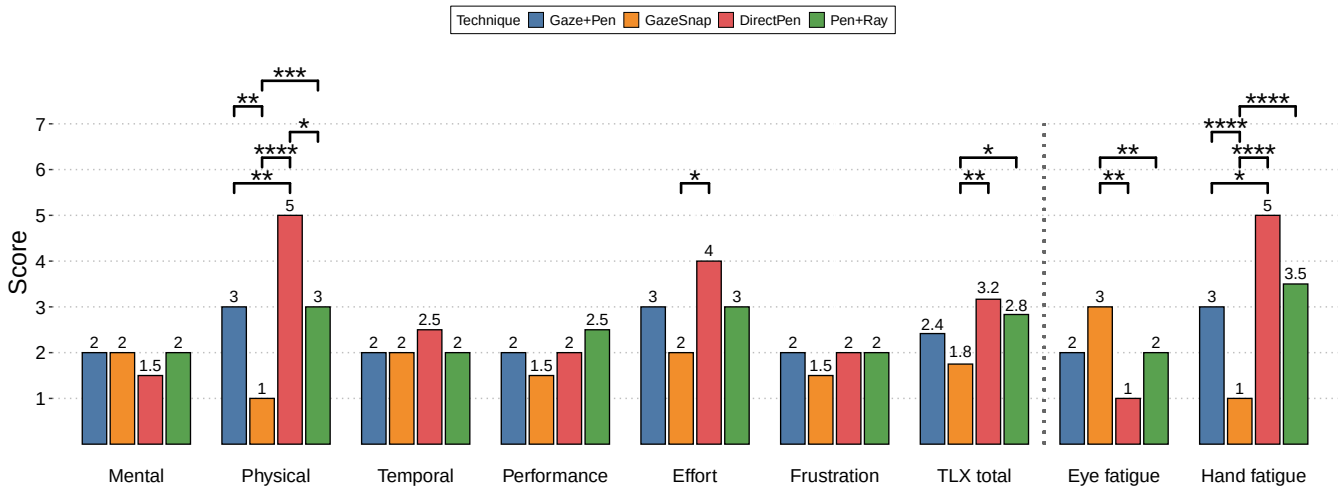
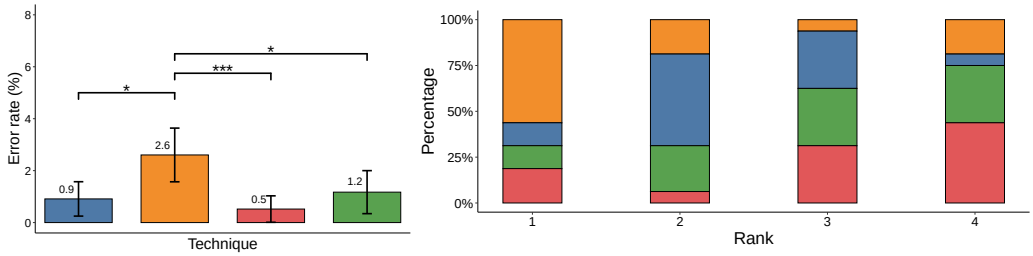


Figure 6: Median NASA-TLXscores showing perceived workload mental, physical, temporal demand, effort, frustration alongside self-reported eye and arm fatigue for each input technique. These results complement the qualitative user feedback presented in Section 5.6.



(a) ER: *TECHNIQUE*.

(b) Preference ranking.

Figure 7: (a) Error Rate for each technique with 95% confidence intervals. (b) Stacked bar chart showing the distribution of subjective preferences ranking for each technique, based on participant response (lower rank = higher preference)

(GazeSnap and Gaze + Pen) with two pen-based techniques (DirectPen and Pen-ray) in terms of performance and user experience. We will now discuss the results in terms of our research questions.

6.1 RQ1: User Performance of Gaze-based to Hand-based Interaction

Our findings show that gaze-based input can effectively support pen-based interactions, especially in contexts that benefit from low-effort, fast input - such as multitasking scenarios or creative applications like drawing interfaces. Among the tested techniques, GazeSnap clearly outperformed the others in terms of speed, efficiency, and reduced physical effort. However, this performance gain came at a cost: GazeSnap also resulted in the highest error rate and significantly increased eye fatigue. However, the trade-off needs to be considered in the context of the actual times. The time saved with GazeSnap was about a whole second, leading to approximately 2.5 seconds for each trial, whereas all other techniques resulted in around 3.5 second task completion times. This represents about a 30% reduction in task time - in contrast to the approximately 1.5 % increase in errors.

The increased error rate observed with GazeSnap may be due to factors such as small target sizes, which made precise fixation difficult. Additionally, the so-called “Early Trigger” problem-where actions are unintentionally triggered before the user intended - likely contributed to inaccurate input. Eye fatigue may also have stemmed from the need for users to consciously control their gaze not just for selection and placement, but also to verify action, increasing visual strain. Possible improvements include enlarging targets, optimizing trigger timing, and providing supportive visual feedback to enhance accuracy and reduce cognitive and visual load. Overall, gaze-based interaction holds promise as a complementary input modality to pen-based techniques.

6.2 RQ2: Gaze + Pen versus GazeSnap

As we included two gaze-based techniques, it is particularly insightful to compare their effectiveness. Gaze + Pen represents the default interaction model used in current XR devices such as the HoloLens 2 or Apple Vision Pro. However, our results showed no significant differences between Gaze + Pen and the other techniques - its overall performance was comparable. However, GazeSnap stood out, offering substantial reductions in both temporal and physical effort for users. One notable finding is that GazeSnap led to higher reported eye fatigue compared to two manual baselines, whereas Gaze + Pen showed no significant differences in this regard. The eye fatigue ratings partially traded with hand fatigue, though. We find that GazeSnap has even less physical fatigue than Gaze + Pen. This suggests that offloading only the object selection task to gaze input - as Gaze + Pen does - still imposes a noticeable level of physical effort, which users seem to perceive more strongly than with GazeSnap.

The increase in eye fatigue with GazeSnap can be attributed to the nature of the interaction: users must actively and consciously control their gaze -not only to select and place objects but also to verify and confirm their actions. This sustained visual concentra-

tion, combined with the need to maintain a stable fixation point, can lead to fatigue of the eyes, especially during extended tasks. Design strategies such as larger targets, optimized trigger timing, or more supportive visual feedback could help to mitigate this trade-off and enhance both accuracy and user comfort.

Overall, our findings suggest that snap outperforms the more common Gaze + Pen model in tasks involving frequent control point dragging, where time and effort reduction are key. However, this advantage is task-dependent-GazeSnap may be less practical when moving objects freely without snapping targets. This reflects a trade-off between limited expressiveness and fast, low-effort rough shape creation.

Compared to prior eye-hand studies using hand gestures, our gaze-and-pen technique shows significant performance improvement, largely thanks to the snapping mechanism. Unlike approaches like Look & Drop [41], which require manual depth control, GazeSnap automatically snaps to target depth, eliminating this effort and outperforming Gaze + Pen. Though gaze is often linked to distant interactions, our results show it can compete effectively with direct input even in near-space tasks.

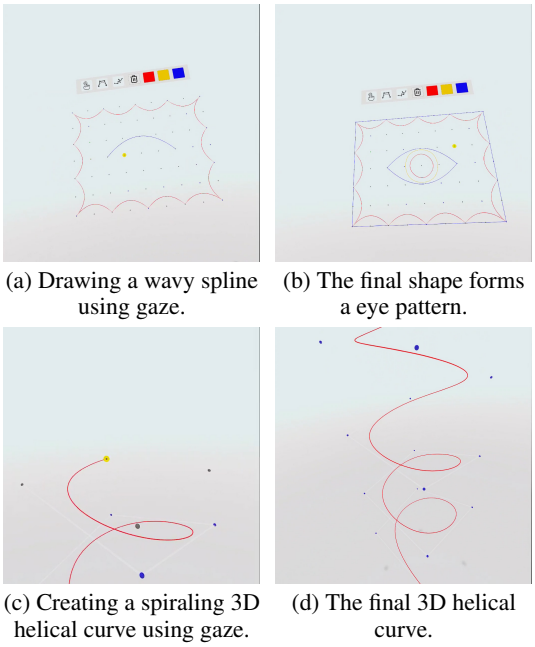
This suggests the potential of using GazeSnap for shape editing tasks, though its application across different shape editing scenarios remains an open area for exploration. Our demos highlight novel spatial interactions with grids of control points—for example, creating lines, curves, and splines based on the selected mode. While these advanced manipulations may require training to master and fully leverage the multimodal control, they demonstrate promising new possibilities for reducing hand/eye fatigue and enhancing spatial design when eye-tracking is more deeply integrated.Beyond the results, we note that there is likely no "best" technique but rather, a future design application can in principle support all techniques so users can take advantage of each techniques strength.

6.3 Limitations

First, we focused on shape point translation tasks with high object movement, which may generalize to other translation tasks but not necessarily to more complex design workflows. Second, while snapping likely improved gaze-based input, it may have also benefited techniques like Pen-ray or DirectPen. Since industrial pens usually use absolute pointing without snapping, including snapping introduced an asymmetry that might have widened the performance gap. Future work should explore pen-based snapping to better quantify its impact and see if GazeSnap's advantage holds under controlled conditions. Third, we tested with target sizes of 4 and 6 degrees visual angle; smaller targets were impractical due to eye-tracking limits. It would also be valuable to assess how well the Quest Pro controller represents natural pen use and to investigate additional parameters for our techniques.

6.4 Application Examples (Figure 8)

We developed a simple, anchor-based creation tool to explore the interaction between gaze and pen, particularly focusing on the practical application of GazeSnap, as its potential use cases may be less immediately apparent as a novel technique. Users can select tools and colors using Gaze + Pen and then create shapes on a grid of anchor points. The tool supports object positioning spline and polyline creation, and control point deletion. **Scenario 1** illustrates how users can select tools, switch colors, and draw 2D shapes by relying on gaze. By targeting anchor points with their gaze, users can switch tools and snap target anchors efficiently. **Scenario 2** demonstrates an advanced interaction where users combine gaze, pen, and pinch gesture with their non-dominant hand (NDH) to create a helical 3D curve. *The pen selects a nearby anchor, the gaze snaps to a distant anchor, and the pinch gesture adjusts the height of the anchor*, forming a classic helix. Both examples showcase the potential of integrating pen and gaze input for more complex creative tasks. The inclusion of hand gestures provides additional control, particularly in 3D design scenarios.



(a) Drawing a wavy spline using gaze.
(b) The final shape forms a eye pattern.
(c) Creating a spiraling 3D helical curve using gaze.
(d) The final 3D helical curve.

Figure 8: Two interaction scenarios using the anchor-based creation tool for multimodal interaction. (a-b): gaze-based selection and 2D spline creation via snapping. (c-d): combined pen, gaze, and pinch gesture input to create a 3D helical curve.

7 Conclusion

This work explored how multimodal gaze and pen input can support point editing in 3D design. GazeSnap consistently enabled faster performance, lower physical effort, and reduced task load, though it came slightly higher error rates and more eye fatigue-revealing a trade-off between efficiency and precision. Still, this trade-off appears acceptable: GazeSnap cut task time by roughly 30%, while errors rose only 1–2%. Most users preferred it, suggesting strong potential for future spatial design tool. Our study focused on point translation as a core shape editing task, serving as a first step toward broader investigations. Future work should explore operations like point insertion, curvature adjustments, or edge modifications to evaluate how input modalities scale to more complex tasks. It would be also valuable to refine targeting and filtering to reduce errors, and to apply these techniques in more advanced workflows-such as full 3D modeling or collaborative design [16]. In collaborative settings, gaze input presents unique challenges due to its low observability compared to hand gestures [21]. Another promising direction is to combine the strength of gaze and pen input - example, using gaze for quick target selection and the pen for interacting with radial or context-sensitive menus. This approach pairs gaze's speed with the pen's precision, especially for tasks needing fine parameter adjustments or mode switches. While this work focused on mid-air pen use, exploring pen-and gaze-interaction on surfaces, which offers different affordances, it also worth pursuing [30].

Acknowledgments

This work was supported by funding from a Google Research Gift award ('Multimodal and Gaze + Gesture Interactions in XR'), the Danish National Research Foundation under the Pioneer Centre for AI in Denmark (DNRF grant P1), and the German Research Foundation (DFG) with project C07 of SFB-TRR 161.

REFERENCES

[1] R. Arora, R. Habib Kazi, T. Grossman, G. Fitzmaurice, and K. Singh. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–15. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173759 2

[2] R. Arora, R. H. Kazi, F. Anderson, T. Grossman, K. Singh, and G. Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 5643–5654. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025474 2

[3] S. Aziz and O. Komogortsev. An assessment of the eye tracking signal quality captured in the hololens 2. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3517031.3529626 4

[4] R. Balakrishnan and G. Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, p. 56–62. Association for Computing Machinery, New York, NY, USA, 1999. doi: 10.1145/302979.302991 4

[5] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300331 2

[6] A. U. Batmaz, A. K. Mutasim, and W. Stuerzlinger. Precision vs. power grip: A comparison of pen grip styles for selection in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 23–28, 2020. doi: 10.1109/VRW50115.2020.00012 3

[7] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *IFIP TC13 International Conference on Human-Computer Interaction*, 2003. 2, 3

[8] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964. 6

[9] P. Brandl, C. Forlines, D. Wigdor, M. Haller, and C. Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, p. 154–161. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1385569.1385595 2

[10] C. Chen, M. Yarmand, Z. Xu, V. Singh, Y. Zhang, and N. Weibel. Investigating input modality and task geometry on precision-first 3d drawing in virtual reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 384–393, 2022. doi: 10.1109/ISMAR55827.2022.00054 3

[11] L. Colligan, H. W. Potts, C. T. Finn, and R. A. Sinkin. Cognitive workload changes for nurses transitioning from a legacy system with paper documentation to a commercial electronic health record. vol. 84, p. 469–476. International journal of medical informatics, 2015. 6

[12] M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving drag-and-drop on wall-size displays. In *Proceedings of Graphics Interface 2005*, GI '05, p. 25–32. Canadian Human-Computer Communications Society, Waterloo, CAN, 2005. 2

[13] M. F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM Trans. Comput.-Hum. Interact.*, 2(3):220–238, Sept. 1995. doi: 10.1145/210079.210087 2

[14] T. Drey, J. Gugenheimer, J. Karlbauer, M. Milo, and E. Rukzio. Vrsketchin: Exploring the design space of pen and tablet interaction for 3d sketching in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376628 2

[15] T. Gesslein, V. Biener, P. Gagel, D. Schneider, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Pen-based interaction with spreadsheets in mobile virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 361–373, 2020. doi: 10.1109/ISMAR50242.2020.00063 2

[16] J. E. S. Grønbæk, J. Sánchez Esquivel, G. Leiva, E. Velloso, H. Gellersen, and K. Pfeuffer. Blended whiteboard: Physicality and reconfigurability in remote mixed reality collaboration. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642293 9

[17] T. Grossman and R. Balakrishnan. 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*, CHI '05, p. 281–290. Association for Computing Machinery, New York, NY, USA, 2005. doi: 10.1145/1054972.1055012 2, 3

[18] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing: a complement to bitmap pointing in guis. In *Proceedings of Graphics Interface 2004*, pp. 9–16, 2004. 2, 3

[19] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, p. 27–36. Association for Computing Machinery, New York, NY, USA, 2010. doi: 10.1145/1866029.1866036 2, 3

[20] A. Hornof, A. Cavender, and R. Hoselton. Eyedraw: a system for drawing pictures with eye movements. *SIGACCESS Access. Comput.*, (77–78):86–93, Sept. 2003. doi: 10.1145/1029014.1028647 3

[21] A. Jing, K. May, B. Matthews, G. Lee, and M. Billinghurst. The impact of sharing gaze behaviours in collaborative mixed reality. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2), Nov. 2022. doi: 10.1145/3555564 9

[22] R. H. Kazi, F. Chevalier, T. Grossman, and G. Fitzmaurice. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pp. 395–405, 2014. 1, 3

[23] R. H. Kazi, F. Chevalier, T. Grossman, and G. Fitzmaurice. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, p. 395–405. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2642918.2647375 2

[24] D. Keefe, R. Zeleznik, and D. Laidlaw. Drawing on air: Input techniques for controlled 3d line illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, 2007. doi: 10.1109/TVCG.2007.1060 2

[25] F. Kern, J. Tschanter, and M. E. Latoschik. Handwriting for text input and the impact of xr displays, surface alignments, and sentence complexities. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 4

[26] C. Latulipe, S. Mann, C. S. Kaplan, and C. L. A. Clarke. symspline: symmetric two-handed spline manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, p. 349–358. Association for Computing Machinery, New York, NY, USA, 2006. doi: 10.1145/1124772.1124825 2

[27] N. Li, T. Han, F. Tian, J. Huang, M. Sun, P. Irani, and J. Alexander. Get a grip: Evaluating grip gestures for vr input using a lightweight pen. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376698 3

[28] M. N. Lystbæk, T. Mikkelsen, R. Krisztandl, E. J. Gonzalez, M. Gonzalez-Franco, H. Gellersen, and K. Pfeuffer. Hands-on, hands-off: Gaze-assisted bimanual 3d interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3654777.3676331 2, 3

[29] A. K. Mutasim, A. U. Batmaz, and W. Stuerzlinger. *Pinch, Click, or Dwell: Comparing Different Selection Techniques for Eye-Gaze-Based Pointing in Virtual Reality*, chap. 15, p. 7. Association for

Computing Machinery, New York, NY, USA, 2021. 1, 2

[30] K. Pfeuffer, J. Alexander, M. K. Chong, Y. Zhang, and H. Gellersen. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, p. 373–383. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807460 1, 3, 9

[31] K. Pfeuffer, J. Alexander, and H. Gellersen. Partially-indirect bimanual input with gaze, pen, and touch for pan, zoom, and ink interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 2845–2856. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858201 3

[32] K. Pfeuffer, A. Dinc, J. Obernolte, R. Rivu, Y. Abdrabou, F. Shelter, Y. Abdelrahman, and F. Alt. Bi-3d: Bi-manual pen-and-touch interaction for 3d manipulation on tablets. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, p. 149–161. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3472749.3474741 2

[33] K. Pfeuffer, B. Mayer, D. Mardanbegi, and H. Gellersen. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, p. 99–108. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3131277.3132180 1, 2, 3, 4

[34] K. Pfeuffer, L. Mecke, S. Delgado Rodriguez, M. Hassib, H. Maier, and F. Alt. Empirical evaluation of gaze-enhanced menus in virtual reality. In *26th ACM Symposium on Virtual Reality Software and Technology*, VRST '20. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3385956.3418962 2, 3

[35] D.-M. Pham and W. Stuerzlinger. Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3359996.3364264 1, 2

[36] L. Qiu, E. S. Kim, S. Suh, L. Sidenmark, and T. Grossman. MaRginalia: Enabling In-person Lecture Capturing and Note-taking Through Mixed Reality, Jan. 2025. arXiv:2501.16010 [cs]. doi: 10.1145/3706598.3714065 3

[37] J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, p. 31–39. Association for Computing Machinery, New York, NY, USA, 1997. doi: 10.1145/263407.263505 2, 3

[38] H. Romat, A. Fender, M. Meier, and C. Holz. Flashpen: A high-fidelity and high-precision multi-surface pen for virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 306–315, 2021. doi: 10.1109/VR50410.2021.00053 2

[39] V. Tanriverdi and R. J. K. Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, p. 265–272. Association for Computing Machinery, New York, NY, USA, 2000. doi: 10.1145/332040.332443 4

[40] R. Turkmen, Z. E. Gelmez, A. U. Batmaz, W. Stuerzlinger, P. Asente, M. Sarac, K. Pfeuffer, and M. D. Barrera Machuca. Eyeguide & eyeconguide: Gaze-based visual guides to improve 3d sketching systems. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3641947 1

[41] U. Wagner, A. Asferg Jacobsen, T. Feuchtner, H. Gellersen, and K. Pfeuffer. Eye-hand movement of objects in near space extended reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3654777.3676446 2, 3, 4, 9

[42] U. Wagner, M. N. Lystbæk, P. Manakhov, J. E. Grønbæk, K. Pfeuffer, and H. Gellersen. A fitts' law study of gaze-hand alignment for selection in 3d user interfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581423 1, 2

[43] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, p. 549–558. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/2047196.2047269 2

[44] S. Wei, D. Bloemers, and A. Rovira. A preliminary study of the eye tracker in the meta quest pro. In *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences*, IMX '23, p. 216–221. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3573381.3596467 4

[45] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, TABLETOP '06, p. 185–192. IEEE Computer Society, USA, 2006. doi: 10.1109/TABLETOP.2006.19 2

[46] K.-P. Yee. Two-handed interaction on a tablet display. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, p. 1493–1496. Association for Computing Machinery, New York, NY, USA, 2004. doi: 10.1145/985921.986098 2

[47] 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*, CHI '99, p. 246–253. Association for Computing Machinery, New York, NY, USA, 1999. doi: 10.1145/302979.303053 3

[48] F. Zhang, K. Katsuragawa, and E. Lank. Conductor: Intersection-based bimanual pointing in augmented and virtual reality. *Proc. ACM Hum.-Comput. Interact.*, 6(ISS), nov 2022. doi: 10.1145/3567713 4

[49] Q. Zou, H. Bai, Z. Chang, Z. Xiao, S. Tian, H. B.-L. Duh, A. Fowler, and M. Billinghurst. The effect of interface types and immersive environments on drawing accuracy and user comfort. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 836–845, 2024. doi: 10.1109/ISMAR62088.2024.00099 2

[50] Q. Zou, H. Bai, L. Gao, G. A. Lee, A. Fowler, and M. B. and. Stylus and gesture asymmetric interaction for fast and precise sketching in virtual reality. *International Journal of Human–Computer Interaction*, 40(23):8124–8141, 2024. doi: 10.1080/10447318.2023.2278294 2