

GazeTune: Facilitating Precise Gaze-Driven Interactions with Cascaded Touch Input

Jina Kim
Graduate School of Culture Technology
KAIST
Daejeon, Republic of Korea
jina1190@kaist.ac.kr

Yang Zhang
Electrical and Computer Engineering
University of California, Los Angeles
Los Angeles, California, USA
yangzhang@ucla.edu

Eric J Gonzalez
Google
Seattle, Washington, USA
ejgonz@google.com

Sang Ho Yoon
Graduate School of Culture Technology
KAIST
Daejeon, Republic of Korea
sangho@kaist.ac.kr

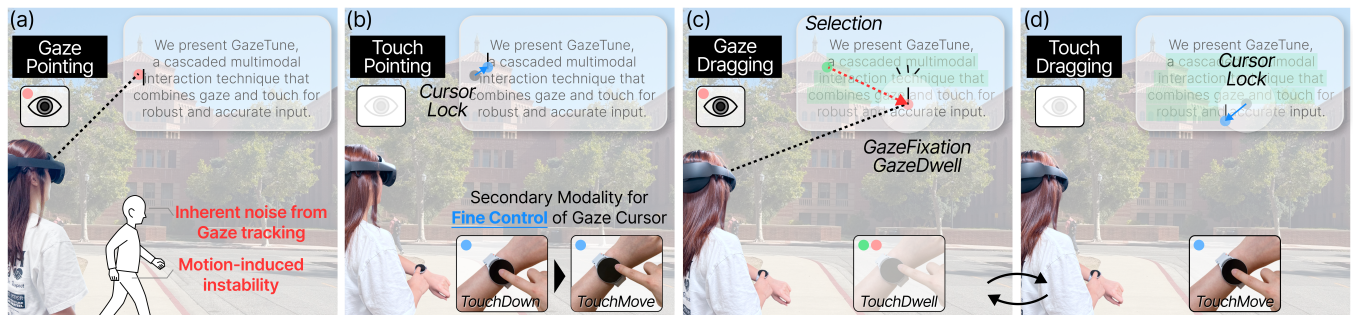


Figure 1: *GazeTune* is a cascaded multimodal interaction combining eye gaze and touch input to overcome natural gaze imprecision and motion-induced instability. Shown here in a text selection workflow: (a) The user coarsely positions the cursor via gaze. (b) Touching the screen locks the cursor, allowing precise refinement via subtle finger movement. (c) To drag an element, the user first dwells their finger to begin selection and then shifts their gaze to the coarse target destination. (d) A new gaze fixation re-locks the cursor for a final touch refinement before finalizing placement.

Abstract

Eye gaze has become an essential input for spatial computing, but its coarse targeting and saccadic nature limit precision and complicate continuous interactions such as dragging, especially under user motion. Gaze+pinch has also become standard in XR for its convenience, yet mid-air gestures remain imprecise, fatiguing, and socially unacceptable. These limitations underscore the need for an approach that preserves the speed of gaze while enabling stable, fine control. We present *GazeTune*, a cascaded multimodal interaction technique combining gaze and touch to refine gaze-based selection and manipulation. Touch serves as a refinement channel within gaze pointing, allowing precise cursor and target control. Our work investigates how gaze-and-touch enhances dragging and mitigates *Motion-Induced* instability. In a study (N=20), we compared *GazeTune* against gaze-only and gaze-pinch methods in 2D dragging. Results show that *GazeTune* achieves significantly lower

error with comparable execution time, validating its effectiveness and balanced trade-off between time and accuracy.

CCS Concepts

• Human-centered computing → Interaction techniques; Mixed / augmented reality.

Keywords

Gaze Interaction, Multimodal Input, Refinement Techniques, Physical Mobility

ACM Reference Format:

Jina Kim, Eric J Gonzalez, Yang Zhang, and Sang Ho Yoon. 2026. GazeTune: Facilitating Precise Gaze-Driven Interactions with Cascaded Touch Input. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26)*, November 02–05, 2026, Detroit, MI, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3830398.3830481>

1 Introduction

Eye gaze has been considered a natural input modality for extended reality (XR), enabling fast and low-effort interaction through a simple glance. Still, gaze input does not support continuous operations such as dragging, which demands sustained precision and stability.



This work is licensed under a Creative Commons Attribution 4.0 International License. *UIST '26, Detroit, MI, USA*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2856-3/26/11

<https://doi.org/10.1145/3830398.3830481>

While gaze is effective for coarse positioning, it lacks the stability needed for fine control due to its saccadic nature and tracking noise. Meanwhile, gaze+pinch interactions [31] have emerged as a convenient and intuitive solution in XR. Yet, mid-air hand gestures remain imprecise, fatiguing, and socially intrusive [14, 27]. As XR devices become increasingly portable (e.g., AR glasses), interactions are expected to function reliably not only in stationary settings but also under locomotion. In such motion-induced conditions, it remains unclear whether gaze+pinch alone can provide stable and precise manipulation. These limitations become particularly evident in dragging, such as adjusting sliders or specifying ranges, where continuous and reliable control is essential. Dragging, therefore, is not merely a stationary interaction but a fundamental operation that extends to use, where precision should be maintained in motion. Rather than extreme movements, this motion refers to everyday urban mobility such as walking or commuting, where users are typically forced to stop due to gait-induced noise when performing such precision tasks.

Several multimodal techniques combine gaze with complementary inputs (e.g., head [22, 35], hand [3, 48], controller [4], touch [2, 38, 42]) to compensate for its inherent imprecision. While effective in stationary settings, these approaches often struggle to support fine-grained corrections during physical movement, as they tend to rely on discrete steps rather than fluid, small-amplitude adjustments. A recent study [48] demonstrated the potential of gaze-assisted dragging with fluid modality transitions, but it did not explicitly address interaction while the user is in motion. Furthermore, as XR moves toward everyday use, users will need precise manipulation techniques that remain practical in public spaces where large, mid-air gestures are undesirable. This highlights the value of leveraging companion devices, like smartwatches, to enable subtle corrective input without adding physical strain or social intrusiveness.

These considerations motivate a gaze-driven design that preserves rapid coarse targeting while enabling fine, continuous manipulation with minimal physical effort. To achieve this, we leverage a smartwatch as a low-overhead, always-accessible complementary channel for subtle touch corrections. Our goal is to combine rapid gaze acquisition with stable touch refinement within a single continuous sequence, reducing physical effort and ensuring reliable control. To this end, we introduce *GazeTune*, a cascaded interaction technique that pairs coarse gaze targeting with precise touch refinement using a commercial smartwatch. The interaction unfolds in four phases: a) setting the coarse start point with gaze, b) locking and refining it via touch, c) specifying a rough endpoint with gaze, and d) refining the final endpoint with touch. Through this flow, gaze offers rapid spatial anchoring while touch enables localized fine-tuning, enabling a natural and reliable dragging experience that neither modality could achieve alone.

To validate our approach, we conducted a user study comparing *GazeTune* against two baselines: *GazePinch* (which represents the state-of-the-art interaction widely used in XR) and *GazeTap* (which captures the inherent limitations of gaze-only dragging). We implemented a Fitts' law-inspired, consecutive selection-and-dragging task across two contexts (*Stationary* vs. *Motion-Induced*) to systematically measure efficiency, accuracy, and workload with a specific focus on user mobility. We found that under the *Motion-Induced* condition, *GazeTune* achieved a significantly lower error

rate (4.24%) compared to *GazePinch* (37.12%) and *GazeTap* (9.70%), while maintaining comparable task completion times. Furthermore, *GazeTune* demonstrated greater stability during manipulation; it yielded faster Drag & Drop times than *GazePinch* (2.59 s vs. 4.09 s) and resulted in a lower dropping error angle (1.44°) than both *GazeTap* (1.89°) and *GazePinch* (2.91°). Overall, our findings demonstrate that *GazeTune* effectively minimizes errors and user workload while preserving the speed advantages of gaze input. This underscores its resilience during user locomotion and validates our seamless modality-switching approach. Overall, these results reveal how users adapt to gaze-and-touch integration, providing insights for future designs that balance rapid targeting with precise correction.

In summary, this work makes three contributions:

- A novel cascaded gaze-and-touch interaction mechanism for precise dragging in XR.
- An empirical evaluation comparing *GazeTune* against existing gaze-driven baselines, validating its ability to enhance interaction precision and stability.
- Design implications for integrating touch input into mobile AR contexts to extend the capabilities of current gaze-based interactions.

2 Related Work

2.1 Gaze-Driven Interaction Techniques

In this subsection, we highlight why refinement is necessary for gaze-driven manipulation. Eye movement-based interaction [16, 41] has become a widely adopted input modality across both 2D devices and immersive XR environments. Because gaze naturally reflects a user's attention and intent to manipulate objects [48], it is often perceived as faster and less physically demanding than other inputs. To further enhance its capabilities, researchers have explored multimodal interactions that combine gaze with additional modalities to improve expressiveness and input accuracy. For example, gaze has been paired with hand gestures [18, 31, 33, 52], as well as touch input for tasks like mode switching [29, 30, 32] and text entry [20, 37]. While these complementary strategies effectively mitigate gaze ambiguity in contexts such as cross-device positioning [45], occluded-object selection [49], depth-aware targeting [4, 54], and voice interaction [55], the majority of these evaluations focus almost exclusively on precise selection.

Consequently, the potential of multimodal gaze systems for continuous operations (e.g., selection and manipulation [46]) - which require sustained adjustments and stable post-selection dragging during physical movement - remains underexplored. We address this gap with *GazeTune* by integrating touch into a cascaded refinement workflow. By pairing natural gaze pointing with subtle touch corrections, our approach enables stable and effective dragging even while the user is in motion.

2.2 Multimodal Refinement for Gaze Input

Even with advanced tracking and calibration [2], gaze alone is often insufficient for precise target selection and remains inherently error-prone [38]. Beyond technical inaccuracies, gaze-based systems suffer from the "Midas touch" problem, which leads to ambiguous or unintended selections. To address this, prior work

Given these limitations, we believe wearable touch offers a practical input channel for precise, selective refinement in mobile contexts. Devices such as smartwatches offer readily available, unobtrusive touch surfaces that integrate into everyday activities without requiring users to hold external hardware or perform large, fatiguing body movements. As such, wearable touch serves as an ideal complement to gaze, providing the stability necessary for fine-grained control on the go.

Prior work has addressed position- and movement-independent gaze interaction on large public displays [19], characterized gaze accuracy and precision ranges on 2D screens [9], and explored continuous eye-movement control via smooth-pursuit adjustment tasks [26]. While gaze has long been considered a powerful input modality, its serious consideration in mobile contexts is relatively recent [25]. Studies have also examined how user movement and task context shape the effectiveness of input. They revealed that locomotion and body motion impair input performance on HMDs across hand-, head-, and eye-based techniques, leading to higher error and latency while increasing attention and motor demands [23, 24, 34]. Furthermore, the necessity of gaze compensation in mobile is highlighted by work showing that the choice of spatial reference critically affects tracking precision [25]. These findings underscore the importance of considering situational impairments but remain focused on discrete acquisition tasks rather than continuous manipulation.

Figure 1: A state transition diagram for the proposed interaction model.

The diagram illustrates the transitions between different interaction states based on user actions. The states are categorized into Gaze actions (red boxes) and Touch actions (blue boxes).

Legend:

- Gaze action**
 - GazeMove**: Continuously exploring the target area
 - GazeFixation**: Gaze on a single location for mode gating
 - GazeDwell**: Maintaining gaze on a point to indicate selection intent
- Touch action**
 - TouchDown**: The initial finger down event on the screen
 - TouchMove**: Continuous finger movement on the screen
 - TouchDwell**: A brief and motionless interval on the screen
 - TouchUp**: The finger leaving the screen

State Transitions:

- Gaze Pointing** (START (Default))
 - Transition 1: Touch Down → **Touch Pointing**
 - Transition 5: Touch Up → **Gaze Dragging**
 - Transition 2: Touch Dwell → **Touch Drugging**
- Gaze Dragging**
 - Transition 4: Gaze&Touch Dwell → **Touch Drugging**
- Touch Pointing**
 - Transition 1: Touch Down → **Gaze Pointing**
 - Transition 2: Touch Dwell → **Touch Drugging**
- Touch Drugging**
 - Transition 2: Touch Dwell → **Touch Pointing**
 - Transition 3: Gaze Fixation (in scope) → **Gaze Dragging**
 - Transition 5: Touch Up → **Gaze Pointing**
- Gaze Fixation (out of scope)**
 - Transition 3: Gaze Fixation (in scope) → **Touch Drugging**
- Gaze Fixation (in scope)**
 - Transition 3: Gaze Fixation (in scope) → **Touch Drugging**

Additional Labels:

- Cursor Lock (Pointing)**: Associated with the initial Touch Down transition.
- Cursor Lock (Dragging)**: Associated with the Gaze&Touch Dwell transition.
- Release without selecting**: Associated with the Touch Up transition from Gaze Pointing.
- Drop Object**: Associated with the Touch Up transition from Gaze Dragging.
- Drop Object**: Associated with the Touch Up transition from Touch Drugging.
- Drop Object**: Associated with the Touch Up transition from Touch Pointing.
- Drop Object**: Associated with the Touch Up transition from Gaze Pointing.

3 GazeTune System Design

In this section, we outline the design goals of *GazeTune*. Our motivation is to enable reliable gaze-driven dragging by combining touch with the rapid targeting advantages of gaze, particularly in scenarios. However, when users are in motion, they cannot maintain a stable device posture or devote sustained visual attention [1]. Under these conditions, our design focuses on enabling low-attention interaction that operates effectively with minimal movement on compact touch surfaces while complementing gaze interaction across a wide XR space. Building on this approach, we pair gaze with subtle touch refinement to support low-effort interaction and natural modality transitions, leading to the following design goals (G):

- [G1] **Leverage Input Familiarity:** To reduce the learning effort, the interaction should align with familiar cursor manipulation metaphors (e.g., trackpad- or mouse-based selection)
- [G2] **Enable Natural Modality Transitions:** Switching between gaze and touch should be seamless to achieve high precision with low cognitive load and minimal physical effort.
- [G3] **Extend Drag Reach from Compact Touch Surfaces:** The technique should remain usable on small touch surfaces typical of wearables, enabling distant selection while minimizing physical clutching.

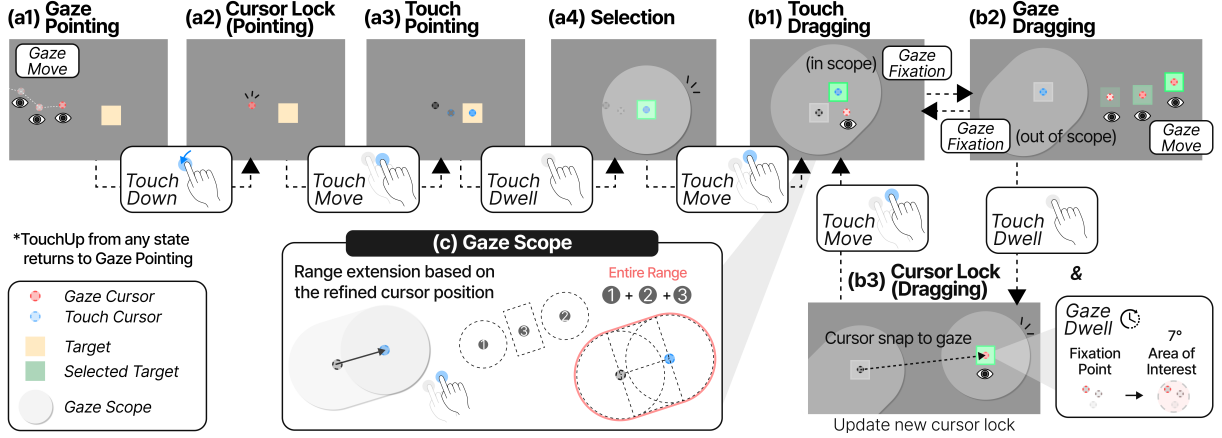


Figure 3: (a1) Gaze pointing to explore the approximate area. (a2) A TouchDown creates a Cursor Lock to anchor the cursor. (a3) Touch pointing to refine the cursor position relative to the lock. (a4) A TouchDwell confirms the selection once the cursor reaches the target. (b1) Touch dragging occurs when the gaze remains within the predefined scope. (b2-3) Gaze dragging occurs when the gaze moves beyond the scope and satisfies fixation-based rules, allowing the cursor to snap to a new location for continued touch dragging. The system can repeat iteratively between two stages based on gaze position. (c) The scope is dynamically updated based on the refined cursor position.

3.2 Overview & Workflow

We designed a cascaded interaction that combines coarse gaze targeting with fine-grained touch control in a single continuous sequence. The *GazeTune* is realized through three gaze actions and four touch actions (Figure 2) for state transitions. Regarding *TouchDwell*, we found that 300 ms was a feasible dwell time, as the touch already indicates intentional use. These events support two linked interaction sequences: Point & Select for target acquisition and Drag & Drop for object manipulation.

- (1) **Gaze Pointing:** As illustrated in Figure 1, the user first explores the approximate area with gaze for coarse target acquisition [21, 43, 44].
- (2) **Touch Pointing:** A *TouchDown* anchors the cursor at the current gaze position. We refer to this anchor as a *Cursor Lock*. The user can *TouchMove* to refine the cursor relative to this lock, enabling subsequent selection and dragging (G1).
- (3) **Select:** When the cursor reaches the target, a *TouchDwell* confirms the selection for subsequent manipulation. For Point & Select, the interaction ends with *TouchUp*.

Our system maintains a *Gaze Scope* around the current lock to regulate transitions. The scope functions as a gating region that determines whether control remains in local touch refinement or transitions to gaze targeting (G2).

- (4) **Touch Dragging:** After selection, the user can *TouchMove* to drag the object while gaze remains within the *Gaze Scope*.
- (5) **Gaze Dragging:** When a larger repositioning is needed, the user shifts gaze outside the *Gaze Scope*. While maintaining a *TouchDwell* and *GazeDwell*, a fixation establishes a cursor lock (dragging) for further refinement where the cursor *snaps* to the current gaze, similar to prior work [42, 52]. If the fixation returns within the *Gaze Scope*, the system returns to *Touch Dragging*.

- (6) **Drop:** The interaction is completed by *TouchUp*, dropping the object at the final location.

By delegating large-scale tasks to gaze and reserving touch for short and anchored refinements, it supports precise local refinement and rapid repositioning within a single sequence that could minimize clutching (G3).

3.3 Cursor Lock for Anchored Refinement

We adopt a *Cursor Lock* mechanism in which gaze provides a spatial anchor while touch enables local cursor refinement. When a *Cursor Lock* is activated, the gaze cursor is immediately frozen, establishing a reference point for subsequent touch-driven movements. This design supports unobtrusive interaction in scenarios by allowing users to refine input quickly without requiring excessive reliance on prolonged visual attention. We define two types of locks, each relevant for a different interaction phase:

- (1) **Pointing Cursor Lock (Touch).** An initial lock is activated during coarse gaze pointing (Figure 3a1) at first touch (Figure 3a2). This locks the cursor and supports local refinement of the pointing cursor before selection (Figure 3a3).

- (2) **Dragging Cursor Lock (Fixation).** Activated during gaze dragging (Figure 3b2). When gaze remains within a 7° radius over a 400 ms window, the system registers stable fixation. And the refinement anchor is re-established at the new fixation point (Figure 3b3).

3.4 Gaze Scope for Mode Gating

To ensure stable refinement and avoid unintended cursor jumps from natural saccadic eye movement, we configure a *Gaze Scope* (Figure 3c), a region surrounding the refinement anchor position. This region determines whether the user intends to perform fine-grained refinement or reposition the cursor over a long distance.

The *Gaze Scope* region dynamically extends based on the refined cursor's position relative to the initial Dragging Cursor Lock

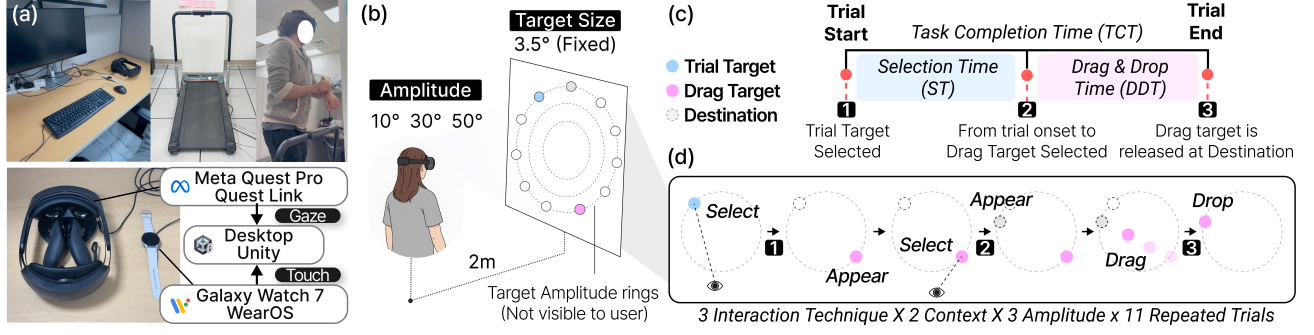


Figure 4: (a) Study setup for the *Stationary* and *Motion-Induced* conditions. Participants walk on the treadmill while wearing an HMD (Meta Quest Pro, 1800 × 1920 pixels per eye, 90 Hz refresh rate) and controlling a smartwatch touchscreen (Galaxy Watch 7, SM-L310, 44 mm). (b) Overall study setup. The trial target appears in a circular layout, and the drag target appears if the user selects it. (c) Task time subcomponents and procedure of each trial. (d) Each trial begins with selecting the trial target. From trial onset to drag target selection is measured as Selection Time (ST). From drag target selection to dropping it at the destination is the Drag & Drop Time (DDT). The total duration is Task Completion Time (TCT). Trials repeat across techniques, contexts, and amplitudes.

anchor. It forms a flexible pill-shaped region (i.e., discorectangle) composed of two 10.1° circles centered on the initial anchor position, the current refinement cursor position, and a rectangular corridor connecting them (Sec 4.3). The Gaze Scope gates the system between two distinct modes of operation during dragging interaction: (1) *Touch Dragging* and (2) *Gaze Dragging*. While the gaze remains inside the Gaze Scope, the system is in a touch dragging mode (Figure 3b1). Gaze is ignored for positional updates, allowing for stable manipulation. When the gaze intentionally moves and remains outside the scope, the system transitions to coarse gaze-based dragging (Figure 3b2). If the user fixates on a new point for over 400 ms, a Cursor Lock (Dragging) “snaps” to the current gaze position, establishing a new anchor for refinement.

This gating mechanism ensures a predictable transition between precise touch-driven manipulation and rapid gaze-assisted aiming. This reduces touchscreen clutching and supports a more seamless and efficient interaction flow.

4 User Study

The objective of our study is to evaluate the performance and explore the behavior of *GazeTune* in comparison to two representative gaze-based techniques. *GazeTap* (Baseline 1) represents the baseline of direct gaze pointing, and *GazePinch* (Baseline 2) is a widely adopted mid-air combination of gaze+pinch, regarded as a commonly used commercial method but prone to fatigue and jitter. We aim to assess both performance and trade-offs among techniques. Specifically, we address following research questions:

- **RQ1:** How does *GazeTune* compare to *GazeTap* and *GazePinch* in overall performance, and what advantages does it provide in the selection and dragging tasks?
- **RQ2:** How does the integration of touch input influence performance under different mobility conditions?
- **RQ3:** How do users adapt their behavior when using *GazeTune*, and how do they perceive its workload and usability?

Table 1: Comparison table for input of each technique. The primary modality for pointing is eye gaze in common.

	GazeTap	GazePinch	GazeTune (Ours)
Select	TouchDown	Pinch	TouchDown
Drag Trigger	TouchDwell	Pinch-in	TouchDwell + GazeDwell
Drag	Eye gaze	Arm Movement	Eye gaze + TouchMove
Drop	TouchUp	Pinch-out	TouchUp

4.1 Study Design Rationale

Smartwatch Use. In line with controller-free XR usage, a smartwatch offers users an instantly accessible body-worn touch surface without retrieving or holding a device for unobtrusive input. Since our technique relies on minimal, low-effort touch to refine gaze-based control, a small anchored surface is more suitable than handheld devices that require explicit grip or posture adjustments. Prior work shows that users naturally stabilize their interacting finger by resting it on the watch, enabling controlled and precise contact gestures [7]. Based on these considerations, we adopted a commercial smartwatch as a practical and reliable source of precise minimal touch input. Despite temporarily engaging both hands, smartwatches offer social and ergonomic advantages, providing discreet interactions and avoiding the fatigue and awkwardness of mid-air gestures. Furthermore, unlike smartphones that require constant holding and retrieval, smartwatches leave hands free.

GazeTap (Touch-based Baseline). *GazeTap* relies on gaze for pointing and dragging, and selection is confirmed by a tap. The user maintains finger contact while the drag position is updated by gaze, and the operation is completed by lifting the finger as illustrated in Table 1. This baseline helps isolate the inherent limitations of gaze-only dragging and motivates the need for touch-based refinement by showing where refinement can contribute to improved control.

GazePinch (Multimodal-based Baseline). *GazePinch* combines gaze pointing with a pinch gesture to trigger and maintain dragging. Once a pinch is performed, the drag position is updated by hand movement until the pinch is released. Although our technique adds

a touch surface, *GazePinch* remains an essential baseline because it is the dominant multimodal interaction on commercial HMDs and the closest deployable alternative for a continuous baseline [48]. It clarifies the benefits of combining gaze with a touch surface and the refinement workflow, providing a practical point of comparison with existing gaze-based indirect arm dragging.

Treadmill Use (Motion-Induced Condition). We also examine how user mobility affects gaze–touch dragging. To validate our method in a physical mobility setting, we included a treadmill *Motion-Induced* condition [24, 34, 50], where body motion can introduce additional gaze offset and stability challenges. We considered two conditions: a *Stationary* condition typical of controlled user studies and a *Motion-Induced* condition where the user walks at a steady pace (1.2 m/s), ensuring a consistent level of body motion across techniques to avoid speed adaptation. We adopted a path-reference [25], which remained at a fixed distance and height to the user’s path of locomotion. We evaluate whether gaze–touch integration can maintain interaction accuracy and speed even under mobility, where gaze input is more susceptible to error.

4.2 Apparatus and Task

Our user study followed ISO 9241-9 [15] for a Fitts’-style evaluation of selection and dragging. Targets were placed on a circular ring at a viewing distance 2 m from the participant and eye height, divided into 11 equal sectors ($\approx 32.7^\circ$). A sector served as the initial index in each block, and subsequent trials are a sector on the opposing side of the previous sector. Target amplitude was manipulated by visual angle with three levels near (10°), mid (30°), and far (50°), while target size was fixed at 3.5° . Each trial followed a three-step sequence. (1) select the trial target to start, which reveals a drag target, (2) select the drag target to reveal the destination target, and (3) drag the drag target to the destination and release it to complete. Participants completed 18 blocks in total, consisting of 3 interaction techniques (*GazeTap*/*GazePinch*/*GazeTune*) $\times$ 2 contexts (*Stationary*/*Motion-Induced*) $\times$ 3 amplitudes (*Near*/*Mid*/*Far*).

We implemented all techniques in Unity 6 using Meta’s built-in Interaction SDK¹ for gaze tracking and pinch detection (Figure 4(a)). Gaze rays were cast onto the task plane and smoothed with a 16 filter to get a stable gaze [4]. Touch events on Galaxy Watch 7² are delivered by *TouchDown*/*TouchMove*/*TouchUp* provided by XDTK [10]. We provided a small semi-transparent gaze cursor and displayed a gaze scope only during refinement. We provided a short haptic pulse for the interaction stage (i.e., when a dragging lock was created) to provide an immediate response to user action using the built-in actuator in watch-enabled conditions (*GazeTap* and *GazeTune*) [5, 12]. We did not include haptic in the *GazePinch* as the pinch action itself provides cutaneous cues.

4.3 Implementation

GazeTap. Gaze performs both acquisition and continuous dragging, while touch gates selection and commitment. On *TouchDown*, gaze selection occurs with a short dwell (300 ms). While the touch is held, the selected target continuously follows the gaze without

anchoring. On *TouchUp*, the dragging ends, and then the stage is cleared.

$$\text{pos}_{\text{target}}(t) = \text{pos}_{\text{gaze}}(t) \quad (1)$$

GazePinch. We used *IndexPinchSelector* from Meta SDK, which provides *WhenSelected*/*WhenUnselected* events to detect pinch-in/out. On *Pinch-in*, the system checks if the gaze lies within the target and, if so, enters the dragging stage and records an initial world-space offset between the target and *OculusCursor* serves as the gaze-anchored cursor proxy. During dragging, the target is moved by *OculusCursor* by controlled by arm movement while preserving the constant offset (Equation 2). On *Pinch-out*, the system exits the dragging stage and finalizes the manipulation. This follows the familiar “gaze to aim, pinch to hold” workflow while providing an indirect drag driven by gaze and hand.

$$\begin{aligned} \text{pos}_{\text{target}}(t) &= \text{pos}_{\text{cursor}}(t) + \Delta_0, \\ \Delta_0 &= \text{pos}_{\text{target}}(t_0) - \text{pos}_{\text{cursor}}(t_0). \end{aligned} \quad (2)$$

GazeTune. On *TouchDown*, the system locks the current gaze position and anchors the cursor. During *TouchMove*, selection is confirmed when the refined cursor enters the target and satisfies the dwell execution time (same as *GazeTap*). For touch dragging, on-screen *TouchMove* deltas directly drive refinement. For gaze dragging, the target snaps toward the current gaze if the gaze leaves the scope while *TouchDwell* remains active. The pill-shaped scope consists of two fixed-radius circles centered at the initial lock and refined cursor, plus the connecting corridor between them. Accordingly, gaze gating is determined by three distances: the distance from the current gaze to the initial anchor, the distance to the current refined cursor, and the perpendicular distance to the connecting segment.

A fixation-based Cursor Lock is performed during a brief *TouchDwell* when three requirements are met: (1) a valid fixation, (2) the gaze is out-of-scope, and (3) low touch speed to avoid accidental recentering. Once a cursor lock is established, refinement resumes from the updated anchor position. On *TouchUp*, the system ends the refinement and clears the lock.

$$\text{Touch Dragging: } \text{pos}_{\text{target}}(t) = \text{pos}_{\text{refined}}(t), \quad (3)$$

$$\text{Gaze Dragging: } \text{pos}_{\text{target}}(t) = \text{pos}_{\text{gaze}}(t). \quad (4)$$

Touch input was mapped to a non-linear control–display gain, where $\Delta \mathbf{p}$ denotes the touch delta in pixel units. We computed the touch velocity as $\mathbf{v} = \Delta \mathbf{p} / \Delta t$ with speed $s = \|\mathbf{v}\|$,

$$g(s) = g_{\min} + (g_{\max} - g_{\min}) \left(1 - e^{-s/v_{\max}}\right),$$

where $g_{\min} = 0.001$, $g_{\max} = 0.003$, and $v_{\max} = 250$. The delta movement is based on smartwatch pixel level, which is scaled by CD gain and mapped to the corresponding visual angles at a 2 m viewing distance. The scope was not a hard boundary on cursor movement but a gating region for switching control modes, remaining controllable through relative gain anchored to the initial cursor lock.

4.4 Participants and Procedure

We carried out an IRB-approved in-person study with 20 participants (8 females, mean age of 24), and 18 participants were right-handed, and 2 participants were left-handed. We first performed calibration for eye gaze and had a practice session to get used to

¹Meta [Interaction SDK](#)

²Samsung [Galaxy Watch 7](#)

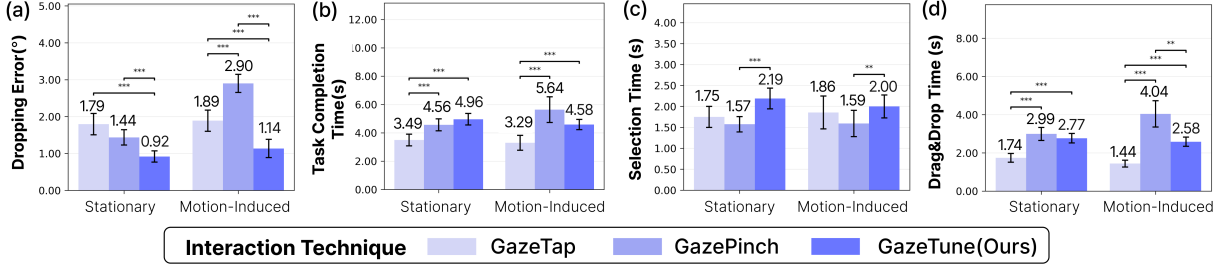


Figure 5: Results across context for each technique with the main effect of technique.

each interaction technique. The total number of conditions was 18 (3 interaction techniques $\times$ 2 conditions $\times$ 3 target amplitudes). We counterbalanced the order of the task blocks for 3 different orders of interaction techniques across participants. Within each block, the context order was blocked (*Stationary* and *Motion-Induced*) to reduce early variability due to motion-induced jitter, and 3 different target amplitude levels were randomized. We collected a total of 3,960 data points (20 users $\times$ 3 interaction techniques $\times$ 2 contexts $\times$ 3 target amplitudes $\times$ 11 trials).

4.5 Measures

Our analysis considers the following metrics with task time decomposed into subcomponents as shown in Figure 4.

- Dropping Error (DE): Angle difference between the ray from the head to the target's final position and the ray from the head to the intended target position at the end of the trial.
- Task Completion Time (TCT): Time from target appearance to dropping.
- Selection Time (ST): Time from target appearance to selection, relevant as follow-up manipulations can impact selection times.
- Drag & Drop Time (DDT): The time from selection to dropping.
- Error Rate (ER): The trial if the target's center is not within the target's radius.
- Subjective Task Load: NASA-TLX (Task Load Index) questionnaire [13] to assess subjective workload and questions about eye and hand fatigue referred to previous work [48].

5 Result

For the quantitative analysis, we first removed outliers using the interquartile range (IQR). We detected that a task completion time exceeds the mean $+ 3 \times$ SD or tracking-induced errors. In total, we removed 130 out of 3,960 trials (3.28%). Then, we checked the normality of the data and applied a log transformation for non-normally distributed factors. We used repeated-measures ANOVA with Greenhouse-Geisser corrections when sphericity was violated. We also performed pairwise comparisons with Bonferroni correction to adjust the p-value. For subjective data, we used a Friedman test with Bonferroni corrected post-hoc Conover tests.

5.1 Performance

5.1.1 Dropping Error (DE). We observed main effects of CONTEXT ($F_{19,00}^{1.00} = 42.47, p < .001$), TECHNIQUE ($F_{36,21}^{1.91} = 64.52, p < .001$), and

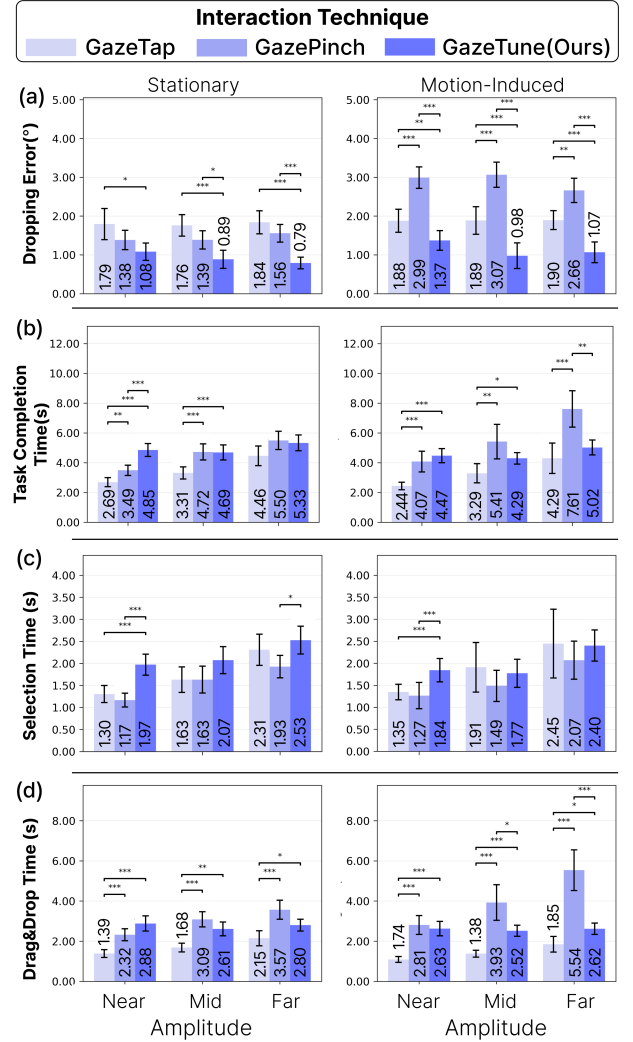


Figure 6: Results across context and amplitude for each technique with the main effect of technique.

CONTEXT $\times$ TECHNIQUE ($F_{28,01}^{1.47} = 32.90, p < .001$). Errors increased under *Motion-Induced* overall, but the pattern was technique-specific. *GazePinch* showed a marked rise (1.44 $\rightarrow$ 2.91, $\Delta = +1.47, p < .001$), *GazeTune* a smaller rise (0.92 $\rightarrow$ 1.14, $\Delta = +0.22, p < .05$), whereas

GazeTap was relatively stable ($1.80 \rightarrow 1.89$, $\Delta = +0.09$). In *Stationary*, *TECHNIQUE* was significant ($F_{33.87}^{1.78} = 25.24$, $p < .001$), while *AMPLITUDE* and the *TECHNIQUE* $\times$ *AMPLITUDE* were not significant. *GazeTune* minimized error across amplitudes. In *Motion-Induced*, *TECHNIQUE* ($F_{32.45}^{1.71} = 66.72$, $p < .001$), *AMPLITUDE* ($F_{34.67}^{1.82} = 4.86$, $p = .016$), and *TECHNIQUE* $\times$ *AMPLITUDE* ($F_{32.91}^{1.73} = 3.71$, $p = .041$) were all significant. *GazeTune* again had the lowest error, and *GazeTap* was lower than *GazePinch*. *Motion-Induced* increased errors and widened technique gaps, with *GazeTune* lower than *GazeTap* and *GazePinch*.

5.1.2 Task Completion Time (TCT). The result showed a main effect of *TECHNIQUE* ($F_{26.60}^{1.40} = 25.68$, $p < .001$), but no main effect of *CONTEXT*. An *CONTEXT* $\times$ *TECHNIQUE* was also observed ($F_{15.82}^{0.83} = 16.03$, $p < .002$). Technique ordering stayed the same, but the gap differed by context. *GazePinch* slowed under *Motion-Induced* ($4.57 \rightarrow 5.70$ s), whereas *GazeTap* and *GazeTune* showed no reliable change. In *Stationary*, both *TECHNIQUE* ($F_{31.38}^{1.65} = 23.33$, $p < .001$) and *AMPLITUDE* ($F_{30.99}^{1.63} = 57.73$, $p < .001$) showed significant main effects with a significant interaction ($F_{38.72}^{2.04} = 7.93$, $p = .001$). Participants were fastest with *GazeTap*, and the time worsened with amplitude. The technique gap changed across amplitudes. In *Motion-Induced*, the same pattern held that *GazeTap* remained fastest, and amplitude led to the slower time. Also, the *TECHNIQUE* ($F_{26.96}^{1.42} = 22.97$, $p < .001$), *AMPLITUDE* ($F_{33.84}^{1.78} = 46.91$, $p < .001$) and *TECHNIQUE* $\times$ *AMPLITUDE* interaction were significant ($F_{29.24}^{1.54} = 11.62$, $p < .001$).

5.1.3 Selection Time (ST). Within Amplitude, we found a main effect of *TECHNIQUE* ($F_{30.89}^{1.62} = 12.96$, $p < .001$), but no main effect of *CONTEXT* and *CONTEXT* $\times$ *TECHNIQUE* were observed. Descriptively, *GazePinch* yielded the shortest selection times ($1.58 \rightarrow 1.61$ s), followed by *GazeTap* ($1.75 \rightarrow 1.90$ s) and *GazeTune* ($2.19 \rightarrow 2.01$ s).

In *Stationary*, the effect of *TECHNIQUE* ($F_{27.06}^{1.42} = 13.64$, $p < .001$), *AMPLITUDE* ($F_{34.31}^{1.81} = 36.24$, $p < .001$) and *TECHNIQUE* $\times$ *AMPLITUDE* ($F_{34.27}^{1.80} = 3.63$, $p = .041$) were significant. *GazeTune* was reliably slower than *GazeTap* and *GazePinch*, and amplitude increased time. In *Motion-Induced*, *TECHNIQUE* ($F_{37.44}^{1.97} = 11.64$, $p < .001$) and *AMPLITUDE* ($F_{32.66}^{1.72} = 56.52$, $p < .001$) were significant but the *TECHNIQUE* $\times$ *AMPLITUDE* was not. The result shows *GazeTune* was slower, and the amplitude increased the selection time. *GazePinch* and *GazeTap* was faster than *GazeTune* in both contexts.

5.1.4 Drag&Drop Time (DDT). Within Amplitude, there was a main effect of *TECHNIQUE* ($F_{22.67}^{1.19} = 46.10$, $p < .001$), and a *CONTEXT* $\times$ *TECHNIQUE* ($F_{13.37}^{0.70} = 20.13$, $p = .001$). The *CONTEXT* was not significant. It showed that *Motion-Induced* reduced DDT for *GazeTap* ($1.74 \rightarrow 1.44$ s, $\Delta = -0.30$ s, $p < .01$) but increased it for *GazePinch* ($2.99 \rightarrow 4.09$ s, $\Delta = +1.10$ s, $p < .001$), with no reliable change for *GazeTune* ($2.77 \rightarrow 2.59$ s). *GazeTap* was faster than *GazePinch* and *GazeTune* in both contexts, with *Motion-Induced* sometimes favoring *GazeTune* over *GazePinch*. In *Stationary*, *TECHNIQUE* ($F_{24.05}^{1.27} = 30.86$, $p < .001$) and *AMPLITUDE* ($F_{34.23}^{1.80} = 24.60$, $p < .001$) were significant. *TECHNIQUE* $\times$ *AMPLITUDE* was also significant ($F_{32.01}^{1.68} = 8.49$, $p = .002$). *GazeTap* was the fastest across amplitudes, and amplitudes make it slower. In *Motion-Induced*, *TECHNIQUE* ($F_{21.95}^{1.16} = 43.62$, $p < .001$), *AMPLITUDE* ($F_{32.09}^{1.69} = 30.41$, $p < .001$), *TECHNIQUE*

Table 2: Drop-error rate by Context $\times$ Technique (DE>3.5°).

Context	Technique	Mean (%)	SD (%)
Stationary	GazeTap	8.94%	12.65%
Stationary	GazePinch	5.91%	6.09%
Stationary	GazeTune	2.88%	2.86%
Motion-Induced	GazeTap	9.70%	14.76%
Motion-Induced	GazePinch	37.12%	15.58%
Motion-Induced	GazeTune	4.24%	4.85%

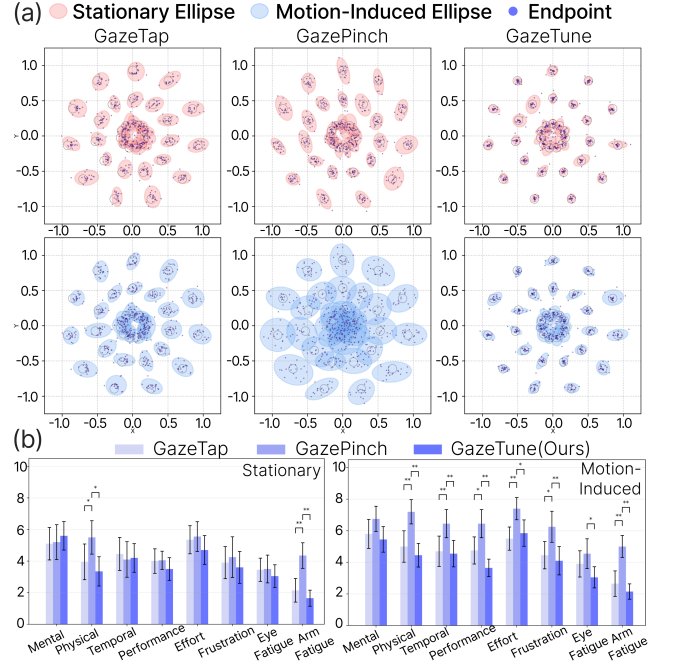


Figure 7: (a) Release point distribution of each technique on each target as a 95% confidence ellipse in visual angle. The removed outlier endpoint was excluded for visualization. (b) NASA-TLX and subject rating of each technique for *Stationary* and *Motion-Induced* conditions.

$\times$ *AMPLITUDE* ($F_{21.99}^{1.16} = 18.67$, $p < .001$) were significant. *GazeTune* sometimes outperformed *GazePinch*.

5.1.5 Error Rate (ER). In *Stationary*, the ordering is *GazeTune* (2.88%) $<$ *GazePinch* (5.91%) $<$ *GazeTap* (8.94%). Under *Motion-Induced*, the pattern was different (*GazeTune* (4.24%) $<$ *GazeTap* (9.70%) $<$ *GazePinch* (37.12%)) and widens markedly due to a large increase for *GazePinch* while *GazeTap* is highest in the seated context. *GazePinch* exhibits a substantial rise with movement, whereas *GazeTune* remains comparatively low with a smaller increase and dispersion.

5.2 Release point for Dragged Target

During drag & drop, the final release is the most vulnerable to motion. To characterize spatial accuracy, we observed the release point distribution with 95% confidence ellipses, which reveals mobility-driven dispersion and whether anchored refinement steadies release.

Table 3: Mean $\pm$ SD of the 95% ellipse size in visual angle across amplitudes, techniques, and contexts.

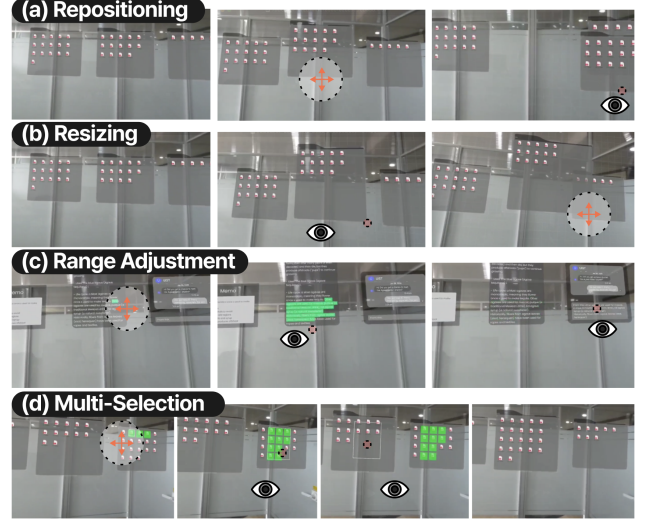
Amplitude	Technique	Stationary		Motion-Induced	
		$\bar{x} (\pm s_x) (^{\circ})$	$\bar{y} (\pm s_y) (^{\circ})$	$\bar{x} (\pm s_x) (^{\circ})$	$\bar{y} (\pm s_y) (^{\circ})$
Near	GazeTap	7.32 $\pm$ 1.22	4.70 $\pm$ 0.70	7.65 $\pm$ 1.29	5.34 $\pm$ 0.98
	GazePinch	6.97 $\pm$ 1.67	4.22 $\pm$ 1.09	12.68 $\pm$ 1.36	9.96 $\pm$ 1.42
	GazeTune	6.11 $\pm$ 1.59	3.23 $\pm$ 0.87	6.99 $\pm$ 0.94	4.63 $\pm$ 1.04
Mid	GazeTap	7.29 $\pm$ 0.77	4.91 $\pm$ 0.75	8.28 $\pm$ 1.04	5.34 $\pm$ 0.98
	GazePinch	7.26 $\pm$ 1.62	4.44 $\pm$ 0.81	13.11 $\pm$ 0.95	10.26 $\pm$ 1.29
	GazeTune	4.97 $\pm$ 1.42	3.43 $\pm$ 1.01	5.96 $\pm$ 0.97	3.07 $\pm$ 0.57
Far	GazeTap	8.37 $\pm$ 0.94	5.80 $\pm$ 0.90	8.60 $\pm$ 1.19	6.44 $\pm$ 1.05
	GazePinch	9.03 $\pm$ 1.52	5.21 $\pm$ 1.22	13.83 $\pm$ 1.65	9.76 $\pm$ 1.78
	GazeTune	4.41 $\pm$ 1.21	3.13 $\pm$ 0.50	6.25 $\pm$ 1.36	3.83 $\pm$ 1.17

Detailed values are reported in table 5.2. First, ellipses are generally larger under *Motion-Induced*, with the gap most pronounced for *GazePinch*. For example, *GazePinch* at Mid is $13.11^{\circ} \times 10.26^{\circ}$ when *Motion-Induced* vs. $7.26^{\circ} \times 4.44^{\circ}$ under *Stationary*. The difference persists at Far (*Motion-Induced* $13.83^{\circ} \times 9.76^{\circ}$ vs. *Stationary*: $9.03^{\circ} \times 5.21^{\circ}$). Moreover, *GazePinch* is the largest, and *GazeTune* is the smallest. Under *Motion-Induced*, *GazePinch* yields the largest dispersion at all amplitudes, while *GazeTune* is consistently the smallest. The same ordering holds for *Stationary* (e.g., *GazeTune* at Far $4.41^{\circ} \times 3.13^{\circ}$). In addition, ellipse sizes generally increase toward Far. For *GazeTap*, ellipse size grows from Near→Mid→Far in both contexts. *GazePinch* increases horizontally with amplitude under *Motion-Induced* while its vertical extent peaks at Mid. Finally, we consistently observe that horizontal exceeds vertical and SD varies. In *Motion-Induced-GazePinch* at Far aspect ratio was 1.42 ($13.83/9.76$). In *Stationary-GazeTune* at Near, it was 1.89 ($6.11/3.23$). This indicates a stronger horizontal error component. Regarding the SD, *Motion-Induced* with *GazePinch* shows the largest variability, whereas *GazeTune* under *Stationary* is comparatively stable.

5.3 Task Load

For *Stationary*, we observed differences mainly in physical load. *GazePinch* is perceived as more demanding on physical demand and arm fatigue, while other factors are comparable. Physical demand differed by technique ($\chi^2(2)=15.44$, $p<.001$, $W=.39$): means were *GazeTune*=3.35 < *GazeTap*=3.95 < *GazePinch*=5.50. Post-hoc tests indicated *GazeTap* < *GazePinch* ($p=.011$) and *GazeTune* < *GazePinch* ($p=.016$), with no difference between *GazeTap* and *GazeTune*. Arm fatigue showed a strong effect ($\chi^2(2)=27.38$, $p<.001$, $W=.68$) with *GazeTune*=1.65 < *GazeTap* (2.15) < *GazePinch* (4.35) and post-hoc revealed *GazeTap* < *GazePinch* ($p=.003$) and *GazeTune* < *GazePinch* ($p<.001$), while *GazeTap* vs. *GazeTune* was not significant.

For *Motion-Induced*, the effects of different techniques had a major impact, reaching significance for *Physical demand* ($\chi^2(2) = 25.14$, $p < .001$, $W = .63$), *Temporal demand* ($\chi^2(2) = 18.38$, $p < .001$, $W = .46$), *Performance* ($\chi^2(2) = 17.22$, $p < .001$, $W = .43$), *Effort* ($\chi^2(2) = 13.00$, $p = .002$, $W = .33$), *Frustration* ($\chi^2(2) = 19.54$, $p < .001$, $W = .49$), *Eye Fatigue* ($\chi^2(2) = 10.78$, $p = .005$, $W = .27$), *Arm Fatigue* ($\chi^2(2) = 25.49$, $p < .001$, $W = .64$) while *Mental demand* was not significant. During *Motion-Induced*, technique

**Figure 8: Use case scenarios of *GazeTune*.**

differences broaden and intensify. A consistent pattern across Effort/Frustration/Physical/Temporal is *GazeTap* < *GazePinch* < *GazeTune* (lower workload for *GazeTap*, highest for *GazeTune*), while Performance favors *GazeTune* best.

5.4 Summary

In this section, we summarize the key findings for each metric. Details of the user feedback are provided in the Appendix A. For Dropping Error, *GazeTune* consistently reduced errors relative to *GazeTap* and *GazePinch* across contexts and amplitudes. While *GazePinch* enabled more accurate drops than *GazeTap* in *Stationary*, *GazePinch* showed a higher error angle in *Motion-Induced*. In comparison, *GazeTune* was more resilient than arm movement dragging in mobile conditions, confirming the achievements of our design goals (precision under mobility). About Task Completion Time, *GazeTap* was the fastest overall, while *GazeTune* became comparable to *GazePinch* under *Motion-Induced* without large time penalties in mobile use. For shorter amplitudes, *GazeTap* was more advantageous, but the difference diminished as the amplitude increased. Regarding Selection Time, *GazeTune* was slower than *GazePinch* at shorter amplitudes. Because *GazeTune* does not require gaze to be precisely within the target, users often place gaze coarsely near the target and then perform additional confirmation drag with dwell phase. For Drag&Drop Time, *GazeTap* was faster than both *GazePinch* and *GazeTune* in *Stationary* and *Motion-Induced* conditions. Notably, *GazeTune* outperformed *GazePinch* in the *Motion-Induced* condition. In Error Rate, *GazeTune* shows the lower error rate across contexts. *GazePinch* can be acceptable in a *Stationary* environment, yet it becomes risky in *Motion-Induced* tasks where drop errors surge.

6 Use Case Scenario

We illustrate how gaze-and-touch supports a unified interaction pattern across near-far tasks that require both rapid navigation and precise control in productivity-oriented AR (Figure 8). We further argue that the technique supports near-far interaction by enabling

seamless transitions between refinement within the gaze scope and rapid repositioning outside the scope. In other words, users can (1) acquire a target or region quickly at a distance, (2) snap the interaction focus to the intended area, and (3) perform stable, fine-grained adjustments within gaze scope. This consistency allows the workflow to generalize across tasks that repeatedly alternate between coarse navigation and precise adjustment.

Repositioning. Users first acquire the target to move with gaze and anchor refinement. While the gaze remains within the scope, they perform small nudges to align the object locally. When the destination lies farther away, they shift their gaze outside the scope to trigger snapping and then re-enter the scope to refine the final placement with touch.

Resizing. Users gaze at and select the resize handle and create an anchor for refinement. Within the scope, refinement supports fine-scale tuning (e.g., matching an edge or fitting a layout). To reach a substantially different size, users look outside the scope to snap the scale toward a new range quickly, then return within the scope for precise final adjustment.

Range adjustment. Users first anchor the start point and refine it within the scope for an exact boundary. To adjust the extent over a larger distance, they shift gaze outside the scope to snap toward a distant candidate endpoint. They then return within the scope to refine the endpoint precisely, enabling iterative tuning by alternating between snapping and local refinement.

Multi-selection. Users anchor an initial item or a boundary point and refine the selection locally within the scope. They then expand coverage by moving the gaze outside the scope to snap the selection window across a larger region and re-enter the scope to refine the boundary again. This supports both precise picking and rapid area coverage.

7 Discussion

Our findings show that combining rapid gaze acquisition with touch-based refinement enables accurate and reliable dragging while maintaining competitive interaction time. A key strength of our approach is the enhanced input stability, achieved by anchoring the cursor to the gaze position and maintaining continuous touch contact. By anchoring the cursor, the user could reduce demanded attention and enable finer and more controlled adjustments. In contrast, mid-air manipulation requires sustained arm elevation, which is unstable and even causes the “Gorilla Arm” effect [17]. These observations indicate that combining gaze acquisition with touch refinement could effectively balance speed and precision in gaze-based interaction.

Balancing Efficiency and Robustness in Mobile Gaze-Driven Interaction (RQ1). While *GazeTune* reduced dropping error, it introduced a trade-off in initial ST, particularly for near targets. This delay inherently stems from the interaction design, which pairs gaze anchoring and touch refinement with a 0.3s dwell confirmation. We observed that even for easy targets (3.5°), users naturally preferred a coarse glance followed by touch adjustment rather than relying on gaze alone. However, this slight increase in ST is outweighed by a reduction in DDT, especially under *Motion-Induced* instability where baseline methods suffered from repeated errors. To mitigate the initial selection delay, our system is designed to

integrate the quick-selection capability of *GazeTap*. Users can fluidly bypass touch refinement when high precision is unnecessary. Under *Motion-Induced* instability and at longer viewing amplitudes, *GazeTune* provides more reliable accuracy while maintaining comparable completion times. This motivates a progressive engagement strategy that begins with rapid gaze targeting and invokes touch refinement when error risk is high (e.g., peripheral placements). Ultimately, this approach benefits spatial computing in large workspaces by maintaining a single, fluid workflow with localized adjustments rather than demanding mode switching.

Cascaded Synergy of Gaze and Touch (RQ2). The consistent performance of *GazeTune* under both *Stationary* and *Motion-Induced* conditions highlights its robust control capabilities. By anchoring the cursor before refinement and re-anchoring for larger repositioning, the interaction reduces reliance on prolonged gaze fixation or sustained mid-air arm movement and ensures stability during body motion. From a motor-control perspective, our cascaded interaction structure includes the ballistic and homing phases of gaze-driven pointing, where touch-based refinement stabilizes the homing phase. This likely helps *GazeTune* maintain comparable performance across near and far targets even under *Motion-Induced* conditions. Furthermore, we found that touch-based refinement offers more practical benefits during continuous post-selection manipulation than during initial target acquisition. The integration of gaze and touch is most effective when their roles are complementary: gaze supports rapid coarse relocation, and touch enables localized stabilization and correction. Together, these properties suggest that *GazeTune* is well-suited for gaze interaction in *Motion-Induced* contexts with a natural gaze-driven workflow.

Behavioral Strategies and Experience of GazeTune (RQ3). Participants adapted quickly to *GazeTune*, although it was initially less intuitive than *GazeTap* and *GazePinch*. This was primarily due to a temporary learning curve associated with the visual split-attention and spatial-mapping coordination required to look at a target in 3D while moving a finger blindly across a 2D wrist-worn interface. However, NASA-TLX results showed no significant negative impact on cognitive demand. As participants adapted easily after a few trials, the initial attentional demand rapidly diminished. Noticeable advantages were observed at the far-edge targets, where small finger adjustments provided more stable placement than head realignment. Interestingly, the average time for each measure in the *Motion-Induced* condition was even slightly faster than in the *Stationary* condition. Users tended to stop refining once the anchored refinement cursor reached a sufficiently accurate region, rather than pursuing marginal improvement. In other words, the touch refinement helps them reach quickly toward the target region, but they simply do not overcorrect. Motion also introduces attention and safety considerations that discourage aggressive micro-corrections, leading participants to end sequences earlier.

Limitations & Future Work. For our evaluation, we limited the mobility condition to ordinary walking speed, as continuous manipulation is rarely required at running speeds. Although evaluation on a treadmill does not fully reproduce the visual motion cues, we intentionally chose this setup to isolate continuous gait-induced noise without self-regulated walking speed. To strengthen generalizability, future work should examine real-world walking scenarios (e.g., circuit-style paths [34]) or outdoor environments

with changing visual backdrop during the movement. Also, our current evaluation lacks a detailed temporal breakdown of the interaction phases. While a DDT comparison with GazeTap (which lacks a refinement phase) provides a rough estimate of one second or less for GazeTune’s manual refinement, this is merely an approximation. Future work should conduct a finer-grained temporal analysis to better understand the temporal overhead of the refinement process. Additionally, our experimental procedure used fixed blocks, so the mobility condition’s order effect was not considered. We also fixed the UI depth at 2 m, which reflects a common spatial UI, but spatial computing systems deploy targets across varied depths. Our results do not guarantee depth-invariant performance. A promising direction is to adapt the gain of refinement to viewing amplitude as depth increases. Additionally, our evaluation focused on a path-anchored interface, whereas spatial interfaces can be anchored to different reference frames (e.g., world-fixed or object-attached). Future work should investigate the refinement behavior under diverse anchoring conditions. In a few cases, users initiated refinement near the edge of the smartwatch, which is a limitation that requires restarting and constrains further adjustment. While our single interaction sequence allowed users to reach most intended positions, edge-case robustness could be further improved by supporting a brief clutching period as an additional gaze-touch gating mechanism that suspends touch until a new gaze anchor is established, allowing safe finger repositioning. Another interesting direction would be to explore surface-based interaction beyond device-bound touch surfaces. For example, on-skin input on the hand or upper arm could provide a larger interaction area with the benefits of surface contact, particularly as sensing technologies continue to improve. Finally, participants suggested system-level enhancements, including increasing the refinement resolution (P2) and adding additional interaction layers or dimensions (e.g., press-based refinements; P19). These extensions could offer richer control while retaining the fluid gaze-driven workflow.

8 Conclusion

In this paper, we introduce *GazeTune*, a cascaded gaze-and-touch technique that supports precise and stable gaze-driven interaction in XR. By offloading coarse acquisition to gaze and limiting touch to anchored refinement, *GazeTune* preserves fast gaze pointing while improving precision. Furthermore, our cascaded design enables interaction within a single continuous sequence to minimize clutching and reduce reliance on prolonged visual fixation. To validate the effectiveness of *GazeTune*, we conducted a comparative study against gaze-only and gaze-pinch baselines. The results indicate that *GazeTune* achieved significantly lower error and maintained consistent dropping accuracy even under *Motion-Induced* conditions with comparable task completion time. In summary, touch input serves as a selective trigger for minimal physical effort, effectively functioning as a precise control interface for selection and drag & drop under physical mobility. We believe our study highlights the potential of cascaded and state-aware input to enable fluid and precise interaction.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea(NRF) grant (RS-2026-25470200, 40%), Institute of Information & communications Technology Planning & Evaluation(IITP) & ITRC(Information Technology Research Center) grant (IITP-2026-RS-2024-00436398, 50%), and the IITP under the virtual convergence support program to nurture the best talents(IITP-2022(2026)-RS-2022-00156435, 10%) grant funded by the Korea government(MSIT).

References

- [1] Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall, and Stuart Tasker. 2003. Multimodal ‘eyes-free’ interaction techniques for wearable devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Ft. Lauderdale Florida USA, 473–480. doi:10.1145/642611.642694
- [2] Zhuojing Cai, Jingkai Hong, Zhimin Wang, and Feng Lu. 2025. GazeSwipe: Enhancing Mobile Touchscreen Reachability through Seamless Gaze and Finger-Swipe Integration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–14. doi:10.1145/3706598.3713739
- [3] 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 (ICMI '15)*. Association for Computing Machinery, New York, NY, USA, 131–138. doi:10.1145/2818346.2820752
- [4] Di Laura Chen, Marcello Giordano, Hrvoje Benko, Tovi Grossman, and Stephanie Santosa. 2023. GazeRayCursor: Facilitating Virtual Reality Target Selection by Blending Gaze and Controller Raycasting. In *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology (VRST '23)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3611659.3615693
- [5] Hyunsung Cho, Naveen Sendhilnathan, Michael Nebeling, Tianyi Wang, Purnima Padmanabhan, Jonathan Browder, David Lindlbauer, Tanya R. Jonker, and Kashyap Todi. 2024. SonoHaptics: An Audio-Haptic Cursor for Gaze-Based Object Selection in XR. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 125, 19 pages. doi:10.1145/3654777.3676384
- [6] Shujie Deng, Nan Jiang, Jian Chang, Shihui Guo, and Jian J. Zhang. 2017. Understanding the impact of multimodal interaction using gaze informed mid-air gesture control in 3D virtual objects manipulation. *International Journal of Human-Computer Studies* 105 (Sept. 2017), 68–80. doi:10.1016/j.ijhcs.2017.04.002
- [7] David Dobbstein, Gabriel Haas, and Enrico Rukzio. 2017. The effects of mobility, encumbrance, and (non-)dominant hand on interaction with smartwatches. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers (Maui, Hawaii) (ISWC '17)*. Association for Computing Machinery, New York, NY, USA, 90–93. doi:10.1145/3123021.3123033
- [8] Ribel Fares, Shaomin Fang, and Oleg Komogortsev. 2013. Can we beat the mouse with MAGIC?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 1387–1390. doi:10.1145/2470654.2466183
- [9] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. 2017. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 1118–1130. doi:10.1145/3025453.3025599
- [10] Eric J. Gonzalez, Khushman Patel, Karan Ahuja, and Mar Gonzalez-Franco. 2024. XDTC: A Cross-Device Toolkit for Input & Interaction in XR. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. Orlando, Florida. <https://github.com/google/xdtk>
- [11] Drew T. Guarnera, Joshua A.C. Behler, Bonita Sharif, and Jonathan I. Maletic. 2025. Automated Fixation Error Correction to Support Eye Tracking Studies on Source Code. *Proc. ACM Hum.-Comput. Interact.* 9, 3 (May 2025), 1–17. doi:10.1145/3725829
- [12] Aakar Gupta, Naveen Sendhilnathan, Jess Hartcher-O’Brien, Evan Pezent, Hrvoje Benko, and Tanya R. Jonker. 2023. Investigating Eyes-away Mid-air Typing in Virtual Reality using Squeeze haptics-based Postural Reinforcement. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 230, 11 pages. doi:10.1145/3544548.3581467
- [13] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (Eds.). Vol. 52. North-Holland, 139–183. doi:10.1016/S0166-4115(08)62386-9
- [14] Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. 2016. Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses. In *Proceedings of the 2016 CHI conference on human factors in*

- computing systems. 4203–4215.
- [15] International Organization for Standardization. 2000. *ISO 9241-9: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 9: Requirements for non-keyboard input devices*. Technical Report ISO 9241-9. International Organization for Standardization.
- [16] Robert J. K. Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*. ACM Press, Seattle, Washington, United States, 11–18. doi:10.1145/97243.97246
- [17] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 3328–3339.
- [18] Jaewon Jeong, Soo-Hyung Kim, Hyung-Jeong Yang, Gun A. Lee, and Seungwon Kim. 2023. GazeHand: A Gaze-Driven Virtual Hand Interface. *IEEE Access* 11 (2023), 133703–133716. doi:10.1109/ACCESS.2023.3337372
- [19] Mohamed Khamis, Axel Hoesl, Alexander Klimczak, Martin Reiss, Florian Alt, and Andreas Bulling. 2017. EyeScout: Active Eye Tracking for Position and Movement Independent Gaze Interaction with Large Public Displays. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, Québec City QC Canada, 155–166. doi:10.1145/3126594.3126630
- [20] Chandan Kumar, Ramin Hedgeshy, I. Scott MacKenzie, and Steffen Staab. 2020. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3313831.3376317
- [21] Manu Kumar, Andreas Paepcke, and Terry Winograd. 2007. EyePoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 421–430. doi:10.1145/1240624.1240692
- [22] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–14. doi:10.1145/3173574.3173655
- [23] Tinghui Li, Eduardo Velloso, Anusha Withana, and Zhanna Sarsenbayeva. 2025. Estimating the Effects of Encumbrance and Walking on Mixed Reality Interaction. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–24. doi:10.1145/3706598.3713492
- [24] Yang Li, Juan Liu, Jin Huang, Yang Zhang, Xiaolan Peng, Yulong Bian, and Feng Tian. 2024. Evaluating the effects of user motion and viewing mode on target selection in augmented reality. *International Journal of Human-Computer Studies* 191 (Nov. 2024), 103327. doi:10.1016/j.ijhcs.2024.103327
- [25] Pavel Manakhov, Ludwig Sidenmark, Ken Pfeuffer, and Hans Gellersen. 2024. Gaze on the Go: Effect of Spatial Reference Frame on Visual Target Acquisition During Physical Locomotion in Extended Reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 373, 16 pages. doi:10.1145/3613904.3642915
- [26] Yafeng Niu, Xin Li, Wenjun Yang, Chengqi Xue, Ningyue Peng, and Tao Jin. 2023. Smooth Pursuit Study on an Eye-Control System for Continuous Variable Adjustment Tasks. *International Journal of Human-Computer Interaction* 39, 1 (Jan. 2023), 23–33. doi:10.1080/10447318.2021.2012979 Publisher: Taylor & Francis _eprint: <https://doi.org/10.1080/10447318.2021.2012979>
- [27] Eduardo GP Palmeira, Alexandre Campos, Igor A Moraes, Alexandre G de Siqueira, and Marcelo GG Ferreira. 2023. Quantifying the 'Gorilla Arm' Effect in a Virtual Reality Text Entry Task via Ray-Casting: A Preliminary Single-Subject Study. In *Proceedings of the 25th Symposium on Virtual and Augmented Reality*. 274–278.
- [28] Khushman Patel, Vrushank Phadnis, Eric J Gonzalez, Hans Gellersen, Ken Pfeuffer, and Mar Gonzalez-Franco. 2025. H2E: Hand, Head, Eye a Multimodal Cascade of Natural Inputs. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 84–89. doi:10.1109/VRW66409.2025.00026
- [29] 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*. ACM, Honolulu Hawaii USA, 509–518. doi:10.1145/2642918.2647397
- [30] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 301–311. doi:10.1145/2984511.2984514
- [31] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*. ACM, Brighton United Kingdom, 99–108. doi:10.1145/3131277.3132180
- [32] Sheikh Rivu, Yasmeen Abdrabou, Thomas Mayer, Ken Pfeuffer, and Florian Alt. 2019. GazeButton: enhancing buttons with eye gaze interactions. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications (Denver, Colorado) (ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 73, 7 pages. doi:10.1145/3317956.3318154
- [33] Rongkai Shi, Yushi Wei, Xueying Qin, Pan Hui, and Hai-Ning Liang. 2023. Exploring Gaze-assisted and Hand-based Region Selection in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* 7, ETRA (May 2023), 1–19. doi:10.1145/3591129
- [34] Yonghwan Shin, Augusto Esteves, and Ian Oakley. 2025. Using Augmented Reality on the go: Understanding the effects of mobility on user performance and subjective workload across eye, head, and hand ray pointing. *International Journal of Human-Computer Studies* 204 (Oct. 2025), 103597. doi:10.1016/j.ijhcs.2025.103597
- [35] Ludwig Sidenmark, Diako Mardanbegi, Argenis Ramirez Gomez, Christopher Clarke, and Hans Gellersen. 2020. BimodalGaze: Seamlessly Refined Pointing with Gaze and Filtered Gestural Head Movement. In *ACM Symposium on Eye Tracking Research and Applications*. ACM, Stuttgart Germany, 1–9. doi:10.1145/3379155.3391312
- [36] Ludwig Sidenmark, Mark Parent, Chi-Hao Wu, Joannes Chan, Michael Glueck, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2022. Weighted Pointer: Error-aware Gaze-based Interaction through Fallback Modalities. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (2022), 3585–3595. doi:10.1109/TVCG.2022.3203096
- [37] Shyamli Sindhvani, Christof Lutteroth, and Gerald Weber. 2019. ReType: Quick Text Editing with Keyboard and Gaze. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. doi:10.1145/3290605.3300433
- [38] Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 2981–2990. doi:10.1145/2207676.2208709
- [39] Sophie Stellmach and Raimund Dachselt. 2013. Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 285–294. doi:10.1145/2470654.2470695
- [40] Mengyuan Sun, Dongliang Guo, Fengyi Yang, and Yapeng Liu. 2025. Enhanced Target Selection in Dense VR Environments Using Single-Hand Frozen Gestures. *International Journal of Human-Computer Interaction* 0, 0 (2025), 1–10. doi:10.1080/10447318.2025.2474473
- [41] Vildan Tanrıverdi and Robert J. K. Jacob. 2000. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, The Hague The Netherlands, 265–272. doi:10.1145/332040.332443
- [42] Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 4179–4188. doi:10.1145/2702123.2702355
- [43] Jayson Turner, Jason Alexander, Andreas Bulling, Dominik Schmidt, and Hans Gellersen. 2013. Eye Pull, Eye Push: Moving Objects between Large Screens and Personal Devices with Gaze and Touch. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer, Berlin, Heidelberg, 170–186. doi:10.1007/978-3-642-40480-1_11
- [44] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. 2013. Eye drop: an interaction concept for gaze-supported point-to-point content transfer. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. ACM, Luleå Sweden, 1–4. doi:10.1145/2541831.2541868
- [45] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. 2014. Cross-device gaze-supported point-to-point content transfer. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, Safety Harbor Florida, 19–26. doi:10.1145/2578153.2578155
- [46] Eduardo Velloso, Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. An Empirical Investigation of Gaze Selection in Mid-Air Gestural 3D Manipulation. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, Cham, 315–330. doi:10.1007/978-3-319-22668-2_25
- [47] Oleg Spakov, Poika Isokoski, and Päivi Majaranta. 2014. Look and lean: accurate head-assisted eye pointing. In *Proceedings of the Symposium on Eye Tracking Research and Applications (Safety Harbor, Florida) (ETRA '14)*. Association for Computing Machinery, New York, NY, USA, 35–42. doi:10.1145/2578153.2578157
- [48] Uta Wagner, Andreas Asferg Jacobsen, Tiare Feuchtnner, Hans Gellersen, and Ken Pfeuffer. 2024. Eye-Hand Movement of Objects in Near Space Extended Reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. ACM, Pittsburgh PA USA, 1–13. doi:10.1145/3654777.3676446
- [49] X. Xu, Y. He, Y. Ge, and Z. Zheng. 2025. EyeExpand: A Low-Burden and Accurate 3D Object Selection Method With Gaze and Raycasting. *Computer Graphics Forum* n/a (2025), e70144. doi:10.1111/cgcf.70144 _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgcf.70144>
- [50] Xiaoying Yang, Xue Wang, Gaofeng Dong, Zihan Yan, Mani Srivastava, Eiji Hayashi, and Yang Zhang. 2023. Header: Sensing Head Gestures for Confirmation

- Dialogs on Smartwatches with Wearable Millimeter-Wave Radar. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 138 (Sept. 2023), 28 pages. doi:10.1145/3610900
- [51] Xin Yi, Leping Qiu, Wenjing Tang, Yehan Fan, Hewu Li, and Yuanchun Shi. 2022. DEEP: 3D Gaze Pointing in Virtual Reality Leveraging Eyelid Movement. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 3, 14 pages. doi:10.1145/3526113.3545673
- [52] Difeng Yu, Xueshi Lu, Rongkai Shi, Hai-Ning Liang, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2021. Gaze-Supported 3D Object Manipulation in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (CHI '21). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3411764.3445343
- [53] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*. ACM Press, Pittsburgh, Pennsylvania, United States, 246–253. doi:10.1145/302979.303053
- [54] Chenyang Zhang, Tiansu Chen, Rohan Russel Nedungadi, Eric Shaffer, and Elahe Soltanaghai. 2023. FocusFlow: Leveraging Focal Depth for Gaze Interaction in Virtual Reality. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (UIST '23 Adjunct). Association for Computing Machinery, New York, NY, USA, 1–4. doi:10.1145/3586182.3615818
- [55] Maozheng Zhao, Henry Huang, Zhi Li, Rui Liu, Wenzhe Cui, Kajal Toshniwal, Ananya Goel, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, Khiem Phi, Shumin Zhai, Iv Ramakrishnan, Fusheng Wang, and Xiaojun Bi. 2022. EyeSayCorrect: Eye Gaze and Voice Based Hands-free Text Correction for Mobile Devices. In *27th International Conference on Intelligent User Interfaces*. ACM, Helsinki Finland, 470–482. doi:10.1145/3490099.3511103

A User Feedback

GazeTap. Participants generally described *GazeTap* as the easiest and fastest technique that required minimal effort. However, many reported frustration when aligning the cursor precisely on targets and expressed lower confidence in the final placement. To compensate, users often resorted to subtle head movements or tensing their eyes for fine alignment. For example, P18 noted, “*Staring at a single target made my eyes tired*,” and P20 added, “*I had to stare and try not to blink, so my eyes got tired*.” Several participants also remarked that while gaze quickly brought the cursor in the right direction, achieving precision was difficult (e.g., P9: “*I often moved my head to line things up*.”).

GazePinch. In a *Stationary*, *GazePinch* was perceived as natural and direct, even for participants without prior AR/VR experience, making it feel intuitive overall. Participants highlighted arm fatigue for *GazePinch*: (P5: “*While it felt straightforward and ‘pretty easy’ when seated, because the simple reach-out made initial acquisition quick. Keeping my arm extended was tiring, and I disliked using it while Motion-Induced*.”) Some also described the repeated reach-out/re-attempt cycles as mentally taxing when they had to correct placement. Most participants found it effortful and reported getting tired more quickly in *Motion-Induced*. Several noted that when strict accuracy was not required, they preferred pinch for its straightforwardness (P7: “*I preferred pinch when precision wasn’t critical*”; P16 echoed this preference).

GazeTune. *GazeTune* was initially less intuitive and took longer than *GazeTap* or *GazePinch*, but most participants reported that it became easy to use after a few trials and was valued for its accuracy. Several emphasized lower attentional demand: rather than holding a rigid gaze on the target, they could refine with a finger while maintaining a softer focus (P3: “*accurate, and I don’t need to focus on the target*”). Participants also appreciated reduced head movement and the ability to make comfortable finger adjustments even while *Motion-Induced* (P2: “*I could keep my head steady and move my*

finger comfortably”). And some participants framed it as a time accuracy trade-off (P6: “*It costs time, but I’m willing to accept that*”. P11: “*The refinement gives high accuracy, but it costs time*”). Once familiar, some found it to be reliably precise, but it took a slightly longer time (P9), while others felt it worked better than pure eye gaze when gaze errors occurred (P20).