Adaptique: Multi-objective and Context-aware Online Adaptation of Selection Techniques in Virtual Reality

Chao-Jung Lai
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
Department of Computer Science
University of California, San Diego
La Jolla, California, USA
chaojunglai@cs.toronto.edu

Mauricio Sousa
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
mauricio@dgp.toronto.edu

Tianyu Zhang
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
Department of Computer Science
University of Rochester
Rochester, New York, USA
tianyuz@cs.toronto.edu

Ludwig Sidenmark
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
lsidenmark@dgp.toronto.edu

Tovi Grossman
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
tovi@dgp.toronto.edu

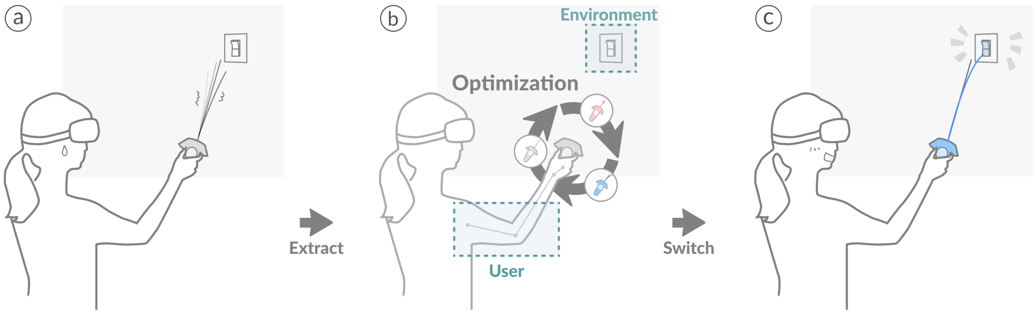

Figure 1: Adaptique switches the selection technique based on environmental and user-based factors, and considers multiple objectives for VR selection. In this example, the user attempts to select the light switch on the far wall to light up the room. (a) Since the light switch is small and far, the user has difficulty selecting it with normal RayCasting. (b) Adaptique continuously senses the environment and user state to find the most optimal selection technique for current use. (c) Adaptique switches the selection technique to StickyRay, snapping the ray to the nearest target to assist the user in accurately and comfortably selecting the light switch.

Abstract

Selection is a fundamental task that is challenging in virtual reality due to issues such as distant and small targets, occlusion, and target-dense environments. Previous research has tackled these challenges through various selection techniques, but complicates selection and can be seen as tedious outside of their designed use case. We present *Adaptique*, an adaptive model that infers and switches to the most optimal selection technique based on user and environmental information. Adaptique considers contextual information such as target size, distance, occlusion, and user posture combined with four objectives: speed, accuracy, comfort, and familiarity which are based on fundamental predictive models of human movement for technique selection. This enables Adaptique to select simple techniques when they are sufficiently efficient and more advanced techniques when necessary. We show that Adaptique is more preferred and performant than single techniques in a user study, and demonstrate Adaptique's versatility in an application.

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; **Virtual reality**; **Interaction techniques**.

Keywords

Virtual/Augmented Reality, Input Techniques, Computational Interaction, Adaptive User Interfaces

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/25/09
https://doi.org/10.1145/3746059.3747790

ACM Reference Format:
Chao-Jung Lai, Mauricio Sousa, Tianyu Zhang, Ludwig Sidenmark, and Tovi Grossman. 2025. Adaptique: Multi-objective and Context-aware Online Adaptation of Selection Techniques in Virtual Reality. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3746059.3747790

1 Introduction

Selection tasks in extended reality (XR) can be challenging in dynamic environments due to factors such as small or distant targets and occlusion [2]. Furthermore, XR environments often change rapidly, with virtual contents changing and users moving or altering their attention within the 3D space [40]. For example, a user might be selecting buttons on a large panel, which requires only a simple and easy selection technique. Then, they might shift to examining components within a complex 3D assembly file they just opened. Since these components are small and cluttered, the user needs a precise selection technique designed to target fine details in a dense environment. Later, they might interact with an IoT light switch on a distant wall to view a real-world object clearly. Because the switch is small and located far away from their reach, they need a technique that can effectively handle distant and small objects. These scenarios represent three distinct environments, making the use of a single technique for all tasks difficult.

Previous works have addressed some of the selection challenges in XR, such as selecting small, distant, and occluded objects, or selection in a dense environment [58, 68] by commonly manipulating the amplitude and width of targets to make pointing and selection easier. However, these techniques are often tailored to specific scenarios and become overly complicated and cumbersome when applied outside their intended context. Manually switching between techniques adds extra workload to the user, who must identify the current context and needs and then perform interaction to switch the technique. To alleviate the burden on users, researchers have proposed adaptive systems that change the current selection technique based on the environment [8, 35] or user factors [57]. However, these works only focus on a single objective for switching or require bespoke scoring algorithms that do not consider the different selection trade-offs (e.g., speed versus accuracy).

We propose *Adaptique*, a novel multi-objective adaptive virtual reality (VR) optimization system that selects the real-time optimal pointing-based selection technique (Figure 1) based on environmental and user factors. In contrast to previous work, Adaptique consists of multiple objectives based on established pointing-based performance metrics such as speed, accuracy, comfort, and familiarity and their associated established models. This enables Adaptique to consider the different trade-offs for selection to holistically select the most optimal technique. Furthermore, using established performance models as optimization objectives ensures reliability and compatibility across different techniques as opposed to bespoke scoring algorithms or machine learning models trained on small data sets. The developer only has to define how a technique affects the model inputs for a given target (i.e., target width and amplitude).

To select the most optimal technique, Adaptique senses the contextual information of selectable targets and the user (Figure 1b) and sends the information to candidate techniques that apply their selection mechanism, which adjust the effective target width and amplitude. The adjusted values are then used as input for the optimization objectives, which calculate the current most effective selection technique for all selectable targets. Adaptique will then make a decision in real-time and switch the technique when the performance reaches the predefined threshold of improvement (Figure 1c). In our current implementation, we included a set of established pointing-based selection techniques of normal RayCasting, StickyRay [23, 42], and RayCursor [3] as candidate selection techniques. These techniques employ distinct mechanisms to modify the effective target size and amplitude compatible with Adaptique and together cover common scenarios of normal selection, small targets, dense environments, and target occlusion.

We showcased Adaptique's utility and applicability in a VR indoor application where Adaptique smoothly switches the selection tool to a more suitable one when the content of the user's interest changes and the task becomes hard with the current tool. Furthermore, our user study highlighted the importance of adaptivity, as using the same technique in different scenarios can lead to difficulty and negatively impact the user experience. We show how Adaptique outperformed the use of singular techniques in selection time, movement, and error rate, and was also preferred by the majority of study participants. In sum, the contributions of this work are:

- Adaptique, a real-time multi-objective adaptive optimization system for selection techniques in VR.
- An application with various selection tasks that showcases the versatility and utility of Adaptique in various natural selection contexts.
- The results of a user study that demonstrate Adaptique benefits, and show how Adaptique outperformed singular techniques in multiple performance metrics and user preference when used across various environments for selection.

2 Related Work

Adaptique builds on common selection challenges in XR, the techniques designed to address these challenges, human selection performance models, and context-aware adaptive systems.

2.1 XR Selection Techniques

RayCasting is one of the common selection techniques in the XR due to its ability to select targets beyond the user's reach by pointing with a ray extending from the user's hand or controller [36, 46]. However, selecting a small or distant object requires higher accuracy because of its small visible area and the tremor of the hand amplified along the ray. In addition, in dense environments, targets may be occluded, resulting in the requirement of physically changing the point of view to be able to see the target. Dense environments also increase the chance of erroneous selection due to the proximity of targets to one another. To address these challenges, various interaction techniques have been proposed.

To enhance the selection of small targets in the 3D space, researchers have proposed techniques that dynamically enlarge the size of objects to expand the interactable area [1], progressive refinement techniques that require steps following the initial action to improve precision [24, 33, 34, 47], snapping mechanisms to decrease the precision requirement by enlarging the effective size [21, 42, 58].

Other works have tackled the problem of selection in dense environments. In addition to the previously mentioned progressive refinement techniques that also help disambiguation in dense environments, some approaches use extra degrees of freedom to specify the depth of the target. For example, Depth Ray [24], RayCursor [3], ClockRay [64], and Alpha Cursor [68] utilize an extra cursor along the ray that is controlled by the forward-backward movement of the hand, swiping on the trackpad, or wrist rotation. MultiFingerBubble [14] uses multiple rays of individual fingers to select between nearby objects by flexing the corresponding finger. Other methods use visual aids, such as mirrors, that display occluded objects from different perspectives and make them visible [37].

Although these techniques make the selection task easier in their designed cases, their standalone use can be overly complex or inefficient when applied outside of their intended context. Adaptique builds on their individual strengths by embedding them within a context-aware adaptive system, allowing each technique to be leveraged where it works best in dynamic XR applications.

2.2 Human Performance Models on Selection Tasks

Researchers have developed various models to evaluate and predict user performance of pointing selection tasks, focusing on speed, accuracy, and comfort. We evaluate these factors using Fitts' Law [19] for speed, the end-point distribution model [67] for accuracy, and the Consumed Endurance model [29] for comfort.

In Fitts' law, the predicted time needed to select a target based on the target's distance and width is formulated as $MT = a + b \cdot log_2(\frac{2A}{W})$ [19]. Here, A represents the amplitude of movement to the target, W is the width of the target along the axis of motion, and the constants a and b are determined by empirical linear regression. The logarithmic term is the index of difficulty (ID) of the task. Though originally applied to 1D selection tasks, it has shown good applicability higher dimension spaces. For example, Shannon formulation defined the movement time as $MT = a + b \cdot log_2(\frac{A}{W} + 1)$. To capture the target geometry, W'-model adjusts the definition of 1D width as the cross-section width along the direction of cursor movement [43]. We adopted this model due to its simplicity, its ability to deal with non-rectangular geometries, and its good fitting result in 2D tasks. In the virtual environment, the Shannon formulation has been used in raycasting tasks because raycasting does not require z-axis movements [21] with two rotation axes as its dominant degree of freedom (DoF) [2]. The target width and amplitude are represented in angular size form to consider the depth [67]. For interaction that requires a higher degree of freedom in translation, such as virtual hand pointing, the 3D Fitts' Law is used more frequently [13, 54].

Endpoint distribution models describe selection behaviors by analyzing the spatial distribution of endpoints during pointing tasks. In XR, models such as the EDModel explore how different factors such as target size, target shape, movement amplitude, and target depth affect the distribution characteristic based on a bivariate Gaussian distribution [67]. Combined with Bi's method [6], this model can estimate selection accuracy by integrating the probability density function for the target region into its control space.

In addition to time and accuracy, user performance is influenced by physical factors such as fatigue and overall comfort. For example, the gorilla arm effect occurs when people feel fatigued in their arms and shoulders after performing mid-air interactions for a long time. Models such as Consumed Endurance (CE) [29] and RULA [45] characterize this ergonomic factor from a biomechanics perspective, relying on physical data such as user postures, arm weights, muscle endurance, and other relevant information. In our work, we utilized these models to evaluate the most suitable selection techniques for the context regarding speed, accuracy, and comfort.

2.3 Adaptive Systems for Interaction Techniques

Recent studies have increasingly highlighted the importance of context-aware adaptive systems in XR interactions [10, 11, 17, 18, 22, 26, 30, 41, 59], especially in mixed reality environments due to their connection to the dynamic physical world. These works have, for example, adapted the layout of virtual content for various kinds of factors, such as the relationship between virtual and physical objects [10, 59], ergonomics [17, 30], physical space [11], or user's intention [22]. These systems are typically implemented through combinatorial optimization, rule-based systems, or data-driven methods such as reinforcement learning. Most XR adaptation systems focus on adapting the layout of virtual content [10, 11, 17, 18, 41], where elements can be freely moved and placed. In contrast, our work assumes that the content and interactable targets in XR are relatively static, as they may represent physical objects that users want to keep intact, with minimal changes to the surrounding environment. Instead, we believe that the user and their interaction should adapt to the current context. Inspired by layout adaptation systems, we employed a multi-objective optimization framework for adaptation due to its simplicity, scalability, and ability to balance multiple objectives in a controllable and interpretable way [50].

There has been limited exploration into adapting selection techniques or selecting appropriate input tools in XR. Although some early efforts have explored the adaptation of selection techniques in virtual environments, these studies either rely on rule-based switching tied to fixed condition, focus on user preference modeling, or remain conceptual without an implemented system [8, 35, 49]. As such, they do not support generalizable, real-time switching based on quantified trade-offs across multiple performance objectives. Other works have adapted the modality of selection techniques based on availability or stability. For example, Sidenmark et al. [57] switched from gaze input to a controller or head-based input when the gaze signal quality dropped, and Yigitbas et al. [65] switched to gaze input when controllers were unavailable. More recently, selection techniques have been included as part of XR layout adaptation systems [11]. In contrast, Adaptique builds on these works in multiple ways: (1) we consider multiple adaptation objectives to reflect selection trade-offs; (2) we use established performance models as optimization objectives to simplify comparability between techniques; and (3) we integrate these models with real-time environmental sensing of selectable candidates and their relationships with each other for online adaptation.

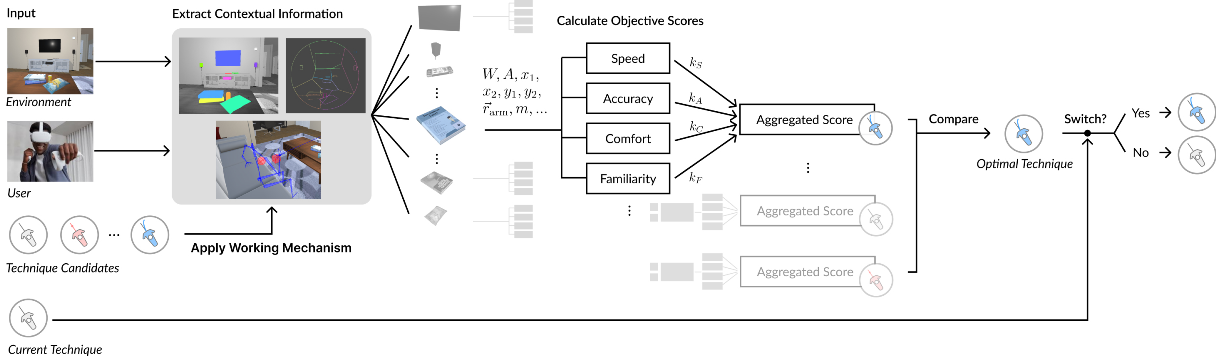

Figure 2: The Adaptique pipeline extracts user and environmental input, applies the working mechanism of selection techniques to calculate model input for each selectable object and technique, aggregates the objects' objective scores for each technique, finds the optimal technique for the interactable objects, and switches the technique if the performance gain is above a threshold.

3 Adaptique

We define the problem of adapting the interaction technique as follows: given a virtual environment with all inferred selection targets, the system will choose the interaction technique that maximizes the performance of the selection task in terms of four objectives: speed, accuracy, comfort, and familiarity. Speed, accuracy, and comfort are three common metrics used to evaluate interaction technique performance [5]. However, while advanced interaction techniques improve performance in selection tasks, they often come with trade-offs such as increased complexity in control and limited interaction expressiveness (e.g., lack of continuous interaction). Overcoming these drawbacks necessitates user familiarity [55]. Therefore, we consider it as one of the inputs in our system. We quantify these metrics and give objective scores to aid in our optimization process. The system would then post-process the data and switch the interaction technique for the user. As illustrated in Figure 2, the system processes each frame by:

(1) Acquire the targets within the interaction space.
(2) Extract contextual information from the user and targets.
(3) Apply the working mechanism of each candidate technique on all selectable targets to use as model objective input.
(4) Calculate and aggregate the objectives for each candidate technique.
(5) Activate a switch if a more optimal one shows a consistent improvement in overall performance.

Sections 3.1–3.3 detail each pipeline component, while Sections 3.4–3.5 explain how specific techniques are integrated and describe system implementation.

3.1 Extracting Contextual Information

We first extract contextual information from the environment and the user that will be used as input for adaptation. Pointing-based selection techniques in XR can be broadly categorized into ray-based techniques, which are effectively 2D techniques in the pointing space, and techniques that use a 3D point, thus incorporating depth information to infer the pointed target [2]. To support both types of techniques, we extract both the 2D and 3D information of selectable targets. We assume that the adaptation should be based only on the user's temporal area of attention, ensuring that only interaction-relevant targets are considered for adaptation and reducing unnecessary computation. Accordingly, we define the interaction space as a cone originating from the pointing direction, with a radius forming an angle of r_c degrees [57]. All objects within this area are included as input for adaptation.

After defining the set of objects for interaction, we extract the most fundamental contextual information to serve as the basis on which techniques apply their working mechanisms to calculate model objective input. This includes the positions and sizes of all targets in both 3D and 2D space relative to the controller. For 3D information, we provide the 3D target positions, shapes, and sizes relative to the controller position. For 2D information, we project all targets' 3D meshes onto a plane perpendicular to the controller's pointing direction. We adopt controller-based projection, as occlusion relative to the controller is more relevant for determining technique applicability, and targets visible to the eyes may still be unreachable given the controller's pose. The projected targets are scaled to ensure that their visual size remains consistent. To incorporate occlusion, we calculate a convex mesh of the projected vertices to form an outline polygon using CGAL [60]. The outline polygon is then clipped by other overlapping polygons that occlude the object using Clipper2 [31]. Finally, we recalculate the target centroid (c) using the final clipped polygon. For each target, we provide the final 2D outline polygon that defines its activation region, along with its position relative to the controller position.

In addition, we incorporate user-side information, including the user's posture and their current selection technique, as the adaptation input. Since most VR systems provide tracking only via controllers and the head-mounted display (HMD), we estimate the user's current posture using inverse kinematics based on the HMD and controller position. Both the 2D and 3D environmental information are updated every frame along with the user information.

3.2 Objectives

To comprehensively consider and balance different interaction goals, Adaptique leverages multiple objectives to find the overall best technique given the contextual information provided. Each objective defines a set of parameters that each interaction technique has to provide for every interactable object to calculate objective scores. Then, all objective scores are calculated for each target and interaction technique. The objective scores of the techniques are then aggregated to a final overall score for each technique.

In our implementation of Adaptique, we use common performance metrics and formalize them with established models of human performance and movement. The system can be expanded to include more objectives and individual objectives can be altered or replaced to better suit the chosen interaction techniques.

3.2.1 *Speed.*

Speed (S_S) is one of the most common performance metrics and selection objectives. We define speed score using the widely adopted index of difficulty (ID) formula in the Shannon formulation of Fitts' Law

$$S_S = -log_2(\frac{A}{W} + 1), \tag{1}$$

which states that the difficulty of selection, and thus the expected selection time, increases with larger movement amplitude and smaller target width [43]. The formula relies on two main parameters, movement amplitude A, and the target size W, both defined in visual angular space(Figure 3a). For this work, we define W as the effective width of the target activation region along the pointing path. The pointing path is defined as the angular trajectory from the current pointing direction to the target centroid, which has been shown to play a greater role than its visual boundaries on selection time [25, 61]. The movement amplitude A is the angular distance the ray needs to travel along that path to the aiming center of the target, which we define as the centroid of the activation region. To ensures that higher performance yields higher scores, we add a negative sign to the objective. Finally, note that Fitts' Law commonly requires the fitting of additional parameters (a and b) based on performance data to predict the selection speed. As all techniques in our implementation are based on controller pointing, we assume that these remain consistent between techniques, thus removing the need for fitting additional parameters and collecting user data. These should be added if Adaptique is expanded to multiple pointing modalities.

3.2.2 *Accuracy.*

Accuracy (S_A) is another critical metric that reflects the reliability of selection. We adopt the *EDModel* by Yu et al. [67]. The EDModel defines an endpoint distribution for pointing-based selection. To calculate the probability of successful selection, we integrate the distribution with the target activation boundary, which we define as our accuracy score:

$$S_A = \iint_D \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) dxdy. \tag{2}$$

The variables μ_x, σ_x, and σ_y are derived via regression using empirical endpoint data [67]. In the integral of Equation 2, we define the x-axis as the direction of movement, the y-axis as perpendicular to the direction of movement, and D as the target activation region. We use the same definition for the direction of movement as in our speed objective. To simplify the integration, we approximate D with a rectangle bounded by coordinates (x_1, x_2, y_1, y_2)–the intersection points of the activation region boundary with the axes (Figure 3b). To derive a closed-form approximation, we leverage the integral identity of the Gaussian distribution over a bounded interval, which can be expressed using the error function (erf). The error function is defined as $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$, representing the cumulative distribution of a standard normal random variable. By substituting the limits of integration corresponding to the boundaries of the activation region D, the double integral in Equation 2 is simplified into a product of one-dimensional error function terms for both the x-axis and y-axis, expressed as:

$$S_A = \frac{1}{4}\left(\text{erf}\left(\frac{y_1}{\sqrt{2}\sigma_y}\right) - \text{erf}\left(\frac{y_2}{\sqrt{2}\sigma_y}\right)\right)\left(\text{erf}\left(\frac{\mu - x_2}{\sqrt{2}\sigma_x}\right) - \text{erf}\left(\frac{\mu - x_1}{\sqrt{2}\sigma_x}\right)\right). \tag{3}$$

3.2.3 *Comfort.*

Comfort plays a vital role in the usability of VR, as physical fatigue, commonly referred to as the gorilla-arm effect [7], can significantly affect user performance. To define comfort (S_C), we adopt a modified Strength metric from the Consumed Endurance (CE) model to quantify arm fatigue caused by selection [29]. In the CE model, the torque exerted on the shoulder must counteract gravitational torque $\vec{g}$:

$$\vec{T}_{shoulder} = ||\vec{r}_{\text{arm}} \times m\vec{g}||. \tag{4}$$

Here, $\vec{r}_{\text{arm}}$ is the distance from the shoulder joint to the center of mass of the arm, and m is the mass of the arm. Since users need to move their arms to reach a target, we calculate the shoulder torque based on predicted user poses during pointing. For simplicity, we assume that users will rotate their forearm toward the aiming center of the target (as defined in subsubsection 3.2.1) at a constant rotational speed, keeping the elbow fixed and aligning the ray with the forearm. While simplified, this approach captures overall fatigue trends and is robust to inaccuracies due to Adaptique's continuous score recalculation as users adjust posture. Furthermore, we know that longer interaction time requires more energy and can lead to fatigue when endurance limits are reached [29]. To quantify this effect of time in energy, we sum shoulder torques from user poses along the pointing trajectory, sampled at β-degree increments from the initial posture (defined in subsection 3.1) to the final target orientation (Figure 3c). This total exertion is then negated to yield the comfort score:

$$S_C = -\sum_{\text{pos}_i} ||\vec{T}_{\text{shoulder,pos}_i}||. \tag{5}$$

3.2.4 *Familiarity.*

Although an advanced technique may be more efficient according to the defined objectives, users may prefer simpler techniques when sufficient to reduce effort and cognitive load. To capture this, each technique is assigned a *"familiarity"* score:

$$S_F = S_{F,\, tech} \text{ if Technique} = tech. \tag{6}$$

This score reflects the added complexity associated with advanced selection techniques and how they may limit interaction in qualitative ways, which is a trade-off between performance and usability. For example, a technique designed for occluded selection may rely on more interaction steps than normal pointing and include mechanisms that alter the typical pointing user experience by, for example, only supporting discrete selection and not continuous targeting [3, 47]. Therefore, simpler techniques that align more closely with traditional pointing are expected to have

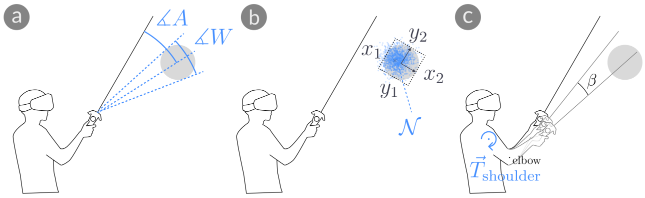

Figure 3: Scoring parameters used for objectives (a) Speed, (b) Accuracy, and (c) Comfort. The grey sphere is the target.

higher familiarity scores than advanced techniques that require more interaction steps [55]. In our implementation, we give each candidate technique a constant value $S_{F,\,tech}$ based on pilot testing in simple selection tasks. In the future, we envision the potential for individual adaptation based on user exposure and performance with each technique [49].

3.2.5 Normalization and Aggregation. To ensure that objective scores are comparable and measured on a consistent scale, we normalize these scores using Min-Max normalization, where the minimum (s_{min}) and maximum (s_{max}) represent the theoretical limits of each objective model. We consider extreme cases in our implementation for limits that do not have a theoretical bound. For example, we consider the smallest target size based on the display limitations and the largest target amplitude as the angle of the interaction region cone (r_c), and the largest possible motion at the most strenuous user position. For special cases such as when an object's activation region is zero due to occlusion, we assign the minimum value as the target is unselectable. To reduce noise caused by environmental or user factors, we apply an exponential smoothing factor to all objectives calculated for each target and technique, defined by a smoothing factor α. Therefore, the score of each objective in the time frame t is defined as

$$S_t = \alpha \times \frac{\sum s_{\text{obj}_i}/N - s_{min}}{s_{max} - s_{min}} + (1-\alpha) \times S_{t-1}. \tag{7}$$

To aggregate the effects of all objects within the interaction space and into a single representative value, we calculate the average of each objective score across all objects within the interaction space. This implies that each object is treated equally important for optimization. In future versions, weighted averages together with target prediction approaches [12, 28] could be deployed to give objects that are more likely to interact with a higher priority.

3.3 Technique Switching

To consider the comprehensive results of the objectives for deciding the most optimal selection technique, we calculate a weighted sum of the scores and select the one with the maximum overall score:

$$\text{Optimal} = \underset{tech}{\text{argmax}}\,(k_S \times S_S + k_A \times S_A + k_C \times S_C + k_F \times S_F)\,. \tag{8}$$

Designers can give the objectives different weightings (k) depending on user tasks or contexts. For example, in a password input task, designers might want to prioritize accuracy and therefore give a higher weight to the accuracy objective.

Finally, to activate a switch in the interaction technique, we ensure that the optimal technique is optimal for n frames within a w-frame window to ignore brief and sudden technique switches. Within these n frames, the difference between the most optimal technique and the current technique must be greater than t_o. Although this introduces a minor delay, we posit that users will be more susceptible to technique switches if the switches are only performed when needed. w, n and t_o can be adjusted to tune the responsiveness and sensitivity of switching. We applied haptic and audio feedback to help users notice technique switches. The controller and ray also change to a unique color for visual feedback. These design choices aim to mitigate distraction and preserve user awareness in an automatic switching system.

3.4 Selection Techniques

A significant number of selection techniques have been proposed for XR [2] and the selection techniques included in the optimization will have a significant impact on the Adaptique user experience. We chose selection techniques based on a set of criteria and assumptions: (1) the techniques should be pointing-based and the working mechanism should alter the target width and amplitude to be compatible with our implemented objectives; (2) the techniques should cover a range of selection scenarios to benefit from adaptation and switching; (3) we should limit the number of techniques to only one per selection scenario to limit the required user training and confusion due to switching between a large set of techniques; and (4) the techniques should not require extra sensing, hardware, or modalities beyond a typical VR controller. Our version of Adaptique is implemented with three controller-based pointing techniques: *RayCasting*, *StickyRay*, and *RayCursor*. These were chosen to cover a wide range of pointing scenarios, from regular pointing to small targets, dense environments, and occlusion. Although our current implementation focuses on these three techniques, Adaptique can be extended to incorporate additional techniques. To integrate these techniques into Adaptique, the selection technique must define W, A and (x_1, x_2, y_1, y_2) for every target in the interaction space. Each technique must also define S_{F_i} for the familiarity objective. In the following section, we introduce each technique included and explain how these variables are derived to support objective computation.

3.4.1 RayCasting. RayCasting represents the most basic pointing technique in which the user points with a ray originating from the controller and the target hit by the ray is highlighted for selection. Due to its simplicity and popularity in 3D interaction, we treat it as a base case for selection. We define A as the angular distance from the current pointing direction to the target center in 2D space (subsection 3.1), and W as the angular width of the object along the pointing direction; x_1 and x_2 denote the entry and exit points of the 2D object along the pointing direction, while y_1 and y_2 correspond to the intersections of the object's outline with a line perpendicular to the pointing direction and passing through the target center (Figure 4a). For completely occluded targets (i.e., those with no visible activation region in the 2D projection due to occlusion) we assign the minimum score. However, users may change their point of view to reveal such targets, and the system reflects these changes as it continuously adjusts in real time during selection. Despite RayCasting being the most widely used technique, users' inherent hand tremors can result in instability in pointing accuracy,

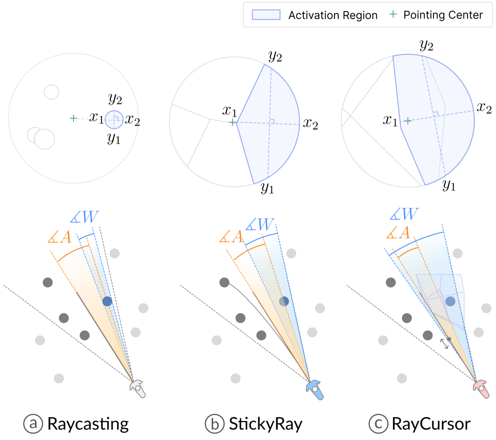

Figure 4: Context score implemented for 3 techniques (a) RayCasting, (b) StickyRay, and (c) RayCursor. The lower figure visualizes the effective size (W) and amplitude (A) of the blue target. The upper figure shows the projection of the activation region of the target.

particularly when selecting small objects. Additionally, when the number of objects in the environment increases, selection becomes more difficult due to the close proximity of objects and occlusion.

3.4.2 StickyRay. We included StickyRay as a second technique for small target selection. StickyRay is based on the Bubble Cursor metaphor [23], where the object nearest to the pointing direction is highlighted for selection. This mechanism expands the effective width of each target to a region that together builds a Voronoi diagram, thus making easy selection of small targets. To show the current closest object, a secondary ray bends toward it [58]. We used the angular distances from the pointing direction to the targets to decide the current closest target, as it has been shown to be the best performing version in 3D settings [42]. As such, the object activation region is the space in which the ray forms the smallest angular distance to the object compared to all other objects (Figure 4b). This is equivalent to computing a Voronoi region in the projected 2D space, where distances are measured by angular distance to the ray. We use Qhull [4] to find the 2D Voronoi region and then clip it by the interaction space. We first find the two intersection points p_1 and p_2 of the line of movement and its projected 2D Voronoi region. Then we get A as the angular distance from the origin to the centroid point of the Voronoi region, and W as the angular distance between p_1 and p_2. x_1, x_2, y_1, and y_2 are defined as with RayCasting but instead using the Voronoi region as target borders.

While it is easier to select small targets compared to RayCasting, StickyRay can be unintuitive, as it encourages pointing outside the visual boundaries of the target. Furthermore, a consequence of the Bubble Cursor mechanism is that a target will always be highlighted for selection, which may not be preferable depending on the context of use. Finally, although StickyRay is proficient in selecting small targets and selection in sparse environments, its benefit diminishes in crowded environments and occluded targets.

3.4.3 RayCursor. To handle dense environments and target occlusion, we included RayCursor, where the user controls a cursor on the ray by swiping on the controller touchpad to select targets at different depths [3]. Like StickyRay, RayCursor has a proximity selection mechanism that will pre-select the object nearest to the cursor to improve performance in selecting small or distant targets. To minimize the need for swiping, the technique has a snapping mechanism that moves the cursor immediately to the depth of the first pointed object's surface. We chose the semi-auto version of RayCursor with the VitLerp transfer function for cursor movement, as it was shown to be the highest performing version [3]. The semi-auto version disables the snapping mechanism when users manually control the cursor through swiping, and reactivates after the trackpad has been released for more than one second. Since RayCursor allows selection either by ray movement alone or in combination with trackpad swiping, its movement is difficult to define. Therefore, we only consider the controller movement for modeling to simplify the calculations. To calculate A and W, we first compute a 3D Voronoi region based on the provided 3D space using QHull [4]. We then project the 3D Voronoi regions to the control space and calculate A and W as in other techniques while ignoring occlusion. x_1, x_2, y_1, and y_2 are defined as with RayCasting and StickyRay but instead using 3D Voronoi projected to control space (Figure 4c).

The RayCursor provides easier selection in dense and occluded environments as it leverages additional depth information for selection. However, the additional interaction steps necessary increase its complexity compared to RayCasting and StickyRay.

3.5 Implementation

We implemented Adaptique in Unity. We used the HTC Vive Pro Eye which has a 110° FOV and a 2880 × 1600 resolution and the Vive controller for pointing input. The controller trackpad was used to control the cursor for RayCursor and the trigger was used to select targets. We used the built-in Unity Inverse Kinematics library to generate user postures. We relied on previous studies to define values for objective parameters that require empirical values. For the accuracy objective, we relied on previous studies by Yu et al. [67] to establish values for the endpoint distribution model: $\mu = -0.1441 \times W + 0.2649$, $\sigma_x = 0.0066 \times A + 0.1025 \times W + 0.2663$, and $\sigma_y = 0.0085 \times A + 0.0679 \times W + 0.1437$. For the comfort objective, we used the equation specified by Hincapié-Ramos et al. [29] to calculate the center of mass for r and m. As for the input data for body parameters we used the average human data for simplicity, specified by Freivalds [20]: 33cm long upper arm weighing 2.1 kg with the center of mass located at 13.2cm; 26.9cm long forearm weighing 1.2 kg with the center of mass located at 11.7cm; 19.1cm long hand weighing 0.4kg with the center of mass located at 7cm.

4 Application

To show Adaptique's performance in dynamic settings, we developed an indoor VR environment where users interact with IOTs, books, food, and UI elements that have different sizes, arrangements, and density. Adaptique is designed to responsively and smoothly switch the selection technique as the user's focus shifts.

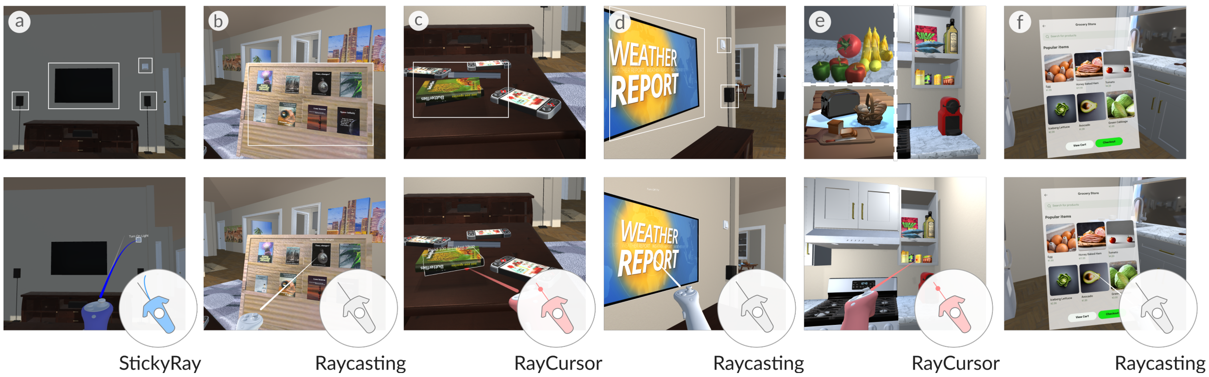

Figure 5: Adaptique chooses different techniques for different scenarios in the application, such as interacting with (a) IOTs on the far wall, (b) books layout on the nearby bookshelf, (c) books stacked on the coffee table, (d) IOTs on the side, (e) ingredients in the cluttered kitchen, and (f) UI panel in front of the users.

In this app, users can point to an interactable target, which displays a brief command description (e.g., 'turn on the light'). By pressing the trigger button on the controller, the command is executed. The application and its interactions are designed to represent common selection scenarios found in VR environments. Adaptique was developed as specified in section 3. We applied the following objective weightings: $k_S = 0.5$ for speed, $k_A = 0.2$ for accuracy, $k_C = 0.15$ for comfort, and $k_F = 0.15$ for familiarity. We applied the following normalized familiarity scores: $S_{F,\ RayCasting} = 0.57$, $S_{F,\ StickyRay} = 0.33$, $S_{F,\ RayCursor} = 0.1$, and the following parameters: interaction space cone with radius $r_c = 20^\circ$, smoothing factor $\alpha = 0.8$, the threshold of $n = 15$ number of frames with improvement above $t_o = 0$ within $w = 20$ windows, and the rotational increments of $\beta = 1^\circ$ to derive interaction postures. These parameters are chosen based on informal pilot testing to ensure the objectives are balanced, the switching is not too sensitive, and no ambiguity on the decision of the most optimal technique.

To exemplify the benefits of Adaptique, we detail a walkthrough of the application (see Video Figure). In the living room of the virtual house, the user first points to the light switch on the wall, attempting to turn on the light. Since it is too far away and small, they struggle to select at the beginning. Adaptique continuously senses the environment and user's state and immediately switches RayCasting to StickyRay. StickyRay snaps the ray to the light switch and makes the selection easier (Figure 5a). Afterward, they want to pick a book to read. There are books laid out on the shelves and stacked on the table. Adaptique smoothly switches to RayCasting when the books are large enough for easy selection (Figure 5b), and switches to RayCursor when the book is occluded (Figure 5c). The user controls the depth of the cursor on the trackpad to pick the book hidden behind. With the responsive assistance of Adaptique, they can precisely select the book to read. They select a sandwich-making guide and decide to go to the kitchen to check the required ingredients. On their way to the kitchen, they turn off the light and TV to save electricity. Adaptique chooses the simplest RayCasting because the IOTs are near and big, thus easy to select (Figure 5d).

The kitchen is cluttered with food and kitchenware. Adaptique chooses RayCursor to handle the dense environment (Figure 5e). This allows them to easily pick the tomato in the stack of fruits, the piece of toast on the cutting board with a blanket of bread around, and the bottle of olive oil arranged on the top cabinet. After they check all the food ingredients, they find that ham and cabbage are out of stock. Therefore, they open an online grocery shopping app to order them. Since the UI buttons on the pop-up panel are designed for easy interaction, Adaptique picks RayCasting for easy interaction (Figure 5f). They select the ingredients icon and the checkout button to order and wait for the ingredients to be delivered. In sum, the application showcases the following advantages:

- Adaptique responsively switches the technique when the content of the user's interest changes, assisting users to interact with a non-homogeneous environment.
- Adaptique comprehensively considers the user's performance in time, accuracy, comfort, and familiarity. When the task is easy enough to be used with a basic technique, Adaptique will stick to the basic one. When the task becomes harder to complete with that technique, it will automatically switch to a more advanced and suitable one.
- Adaptique provides a smooth, consistent, and non-distracting transition by proactively switching the selection tool before users point toward new targets and ensuring that the tool remains consistent when engaging with nearby objects.

5 Evaluation

We conducted a VR user study to explore and evaluate the potential quantitative benefits of Adaptique across different scenarios, compared to using individual techniques. We hypothesized that relying on a single selection technique would introduce performance trade-offs whereas Adaptique always achieve optimal performance.

The study was done through a controlled task where participants were instructed to select one target among many distraction targets as quickly and accurately as possible in different environments. We collected performance metrics, including selection time, error rate, translational movement, and rotational movement. These correspond to our core performance objectives: selection time reflects speed, error rate reflects accuracy, and movement reflects ergonomic cost. To test our hypothesis, we evaluated Adaptique using StickyRay and RayCursor as selection techniques. Raycasting was not included in this performance-based study because, theoretically, its effective size is a lower bound and its effective amplitude is an upper bound, making it inherently outperformed by other techniques on performance metrics [3, 42], and thus unlikely to offer additional insight into the performance trade-offs we aimed to evaluate.

5.1 Task

Participants were tasked to select a target object amongst many distraction targets. We varied the size of the selectable target, the number of targets, and the density of the environment. Target sizes were specified in visual angles, with two conditions: large (2.5°) and small (0.5°), and four different environments that varied in distractors amount and density to cover both simple and extreme cases that users might encounter in XR. The selection target was a sphere and randomly placed within the target region but had to be at least 0.4 meters away from the boundary so that the target was not at the edge within the environment and 0.2 meters away from the center to ensure movement before selection. The distractors were primitive shapes (cubes, spheres, cylinders, and capsules), in pseudo-random positions and sizes (2-4°), and random rotations. They were semitransparent to minimize the effect of visual search in dense and occluded settings and were not intersecting with each other. The environments were also balanced, so that two environment types exploited the advantages of StickyRay, and similarly two environments exploited the advantages of RayCursor. Specifically, the environments were as follows:

Sparse: In SPARSE environment (Figure 6a), the target object and distractors were spawned within a $3m \times 3m \times 3m$ cubic region 2 meters away from the participant. There were a total of 10 objects: 1 target object and 9 distractors. The environment represented a simple case of selection.

Dense: The DENSE environment (Figure 6b) also consisted of a $3m \times 3m \times 3m$ cubic region 2 meters away but contained 240 objects, making the selection target densely surrounded by other objects. The target was likely to be partly occluded by distractors from the view of the participant.

Flat: In FLAT environment (Figure 6c), the spawning region was a $3m \times 3m \times 1m$ cubic region 2 meters away, resulting in a spread-out placement at a similar depth. We used a total of 30 objects.

Deep: In DEEP environment (Figure 6d), the spawning region was a $1.5m \times 1.5m \times 4m$ cubic region 2 meters away. A total of 30 objects were spawned. Though the density of the environment (30 objects in $90m^3$ volume) was the same as FLAT environment, the arrangement of objects extended more in the depth direction.

We pre-generated eight trials of each unique combination of target size and environment to use for all techniques. In sum, the study used the following independent variables and levels:

- TECHNIQUE: STICKYRAY, RAYCURSOR, ADAPTIQUE
- TARGET SIZE: SMALL (0.5°), LARGE (2.5°)
- TARGET ENVIRONMENT: SPARSE, DENSE, FLAT, DEEP

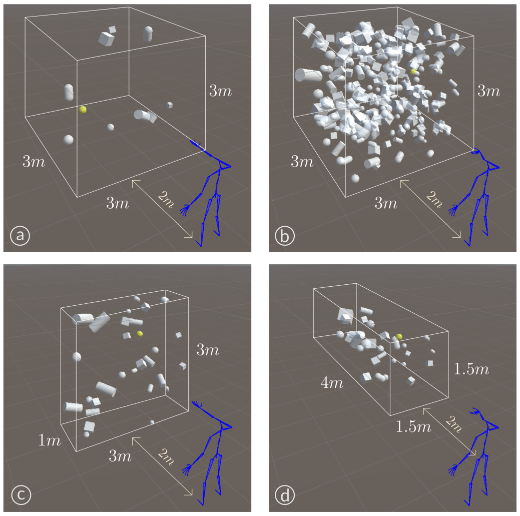

Figure 6: Environments used in the study, including (a) Sparse (b) Dense (c) Flat, and (d) Deep.

5.2 Procedure

Upon arrival, participants completed a consent form and a demographic questionnaire to gather information about participant age, gender, and VR/AR experience before being informed about the study. They were then positioned in a standing posture, equipped with the HMD and controller, and performed a practice session to familiarize themselves with each selection technique before starting the study session. During the study session, the participants performed all selections with a single technique before moving on to the next technique. For each technique, a total of 64 trials (2 TARGET SIZE × 4 TARGET ENVIRONMENT × 8 repetitions) were presented in random order. The order of the techniques was counterbalanced with a balanced Latin square. For each trial, participants needed to select a central "ready" panel before starting the trial. This served as a rest period and ensured that users started the next trial from a central position. Participants were then tasked to select the target object that was highlighted in yellow as quickly and accurately as possible. The participants were unable to move on to the next trial until the correct object had been selected or a 15-second timeout had elapsed. After finishing all the trials with a technique, participants completed a questionnaire consisting of Likert-scale usability questions to capture their experiences. Study usability questions can be found in the supplemental material. Participants then rested before moving on to the next technique. The study was concluded with a questionnaire for user preferences and feedback. In total, we collected 18 participants × 3 TECHNIQUE × 2 TARGET SIZE × 4 TARGET ENVIRONMENT × 8 repetitions = 3456 selections.

5.3 Apparatus and Participants

The interaction techniques and Adaptique were implemented as described in section 3. The participants performed the tasks with the controller in their dominant hand. Selection was done with the trigger button, and depth cursor control with the trackpad. Since our study focuses on performance, we applied the following objective weightings: $k_S = 0.5$, $k_A = 0.2$, $k_C = 0.2$, and $k_F = 0.1$. We applied the following normalized familiarity scores: $F_{\text{StickyRay}} = 0.7$ and $F_{\text{RayCursor}} = 0.3$ based on pilot testing. The rest of the parameters are the same as those in section 4. We recruited 18 participants on campus for the study (12 male, 6 female, 19-32 years old). One used VR/AR weekly, thirteen used VR/AR occasionally, and four had never experienced VR/AR before.

5.4 Results

Unless otherwise stated, the analysis was performed with a 3-way repeated measures ANOVA (α=.05) with Technique, Size, and Environment as independent variables. Before analysis, we removed outlier trials. Trials were discarded if their selection times, translational movement, or rotational movement were beyond 3 standard deviations from their respective grand mean. In total, 164 out of 3456 trials were discarded (4.7%). We tested normality with the Kolmogorov-Smirnov test and QQ-plots. If extreme outliers were identified within the aggregated analysis data, defined as values beyond $Q_R \pm 3 \times$ IQR, a winsorization process was applied. When the assumption of sphericity was violated, as tested with Mauchly's test, Greenhouse-Geisser corrected values were used in the analysis. Bonferroni-corrected post hoc tests were used when applicable. The effect sizes were reported as partial eta squared (η_p^2). Questionnaires were analyzed using Friedman tests, and Bonferroni-corrected Wilcoxon signed-rank tests for post hoc analysis.

5.4.1 Selection Time. We defined the selection time as the time elapsed from the start of the trial to the completion of the selection. We applied a square-root transformation since the distribution of selection time was slightly positively skewed. Significant main effects were observed for Technique ($F_{2,34}$=45.27, p<.001, η_p^2=.73), Size ($F_{1,17}$=72.83, p<.001, η_p^2=.81), and Environment ($F_{1.74,29.50}$=374.63, p<.001, η_p^2=.96). Post hoc analysis of Technique main effect (Figure 7a) revealed that both Adaptique and StickyRay were faster than RayCursor (both p<.001).

Additionally, we found no significant three-way interaction. However, we found significant two-way interactions for Technique × Environment ($F_{3.8,64.60}$=15.02, p<.001, η_p^2=.47) and Technique × Size ($F_{2,34}$=16.58, p<.001, η_p^2=.494). Post hoc analysis of Technique × Environment results (Figure 7b) showed that in the Sparse environment, Adaptique and StickyRay outperformed RayCursor in speed (both p<.001). In the Flat environment, StickyRay emerged as the significantly fastest technique, followed by Adaptique, with RayCursor being the slowest (all p<=.003). In the Deep environment, StickyRay also proved to be faster than RayCursor (p=.032). The techniques did not differ significantly in the Dense environment. For all techniques, users were significantly quickest in the Sparse environment, followed by the Flat, Deep, and Dense environments, the latter resulting in the slowest performance (all p<.001). For the Technique × Size interaction (Figure 7c), both Large and Small targets were selected significantly faster with StickyRay and Adaptique compared to RayCursor (all p<.001).

5.4.2 Movement. We considered translational and rotational movement for the analysis, defined as the total distance traveled and the angle of rotation of the controller from the start of the trial until the selection was completed. Since both metrics were severely positively skewed, we performed a reciprocal transformation to meet the normality requirement.

For translational movement, there were significant main effects of Technique ($F_{2,34}$=29.48, p<.001, η_p^2=.63), Size ($F_{1,17}$=32.81, p<.001, η_p^2=.66), and Environment ($F_{2.05,34.87}$=53.85, p<.001, η_p^2=.76). Adaptique and RayCursor required significantly less movement compared to StickyRay (both p<.001, Figure 7d). Furthermore, no significant three-way interaction was found. Significant two-way interactions were observed for Technique × Environment ($F_{3.68,62.62}$=13.54, p<.001, η_p^2=.46) and Technique × Size. Post hoc analysis of Technique × Environment (Figure 7e) showed that in the Sparse, Dense and Deep environments, Adaptique and RayCursor required significantly less movement compared to StickyRay (all p<=.008). While in the Flat environment, we observed that only Adaptique required significantly less movement than StickyRay (p=.003). For Technique × Size interaction (Figure 7f), selecting both Large and Small targets with Adaptique and RayCursor required less movement than selecting with StickyRay (all p<.001).

Similarly, for rotational movement, the main effects were also significant for Technique ($F_{2,34}$=6.42, p=.004, η_p^2=.27), Size ($F_{1,17}$=54.03, p<.001, η_p^2=.76), and Environment ($F_{1.75,29.81}$=43.69, p<.001, η_p^2=.72). Technique post hoc analysis (Figure 7g) showed that Adaptique again had an advantage, requiring significantly less rotational movement than StickyRay overall (p=.007). However, no significant three-way interaction was found. Significant two-way interactions were found for Technique × Environment ($F_{6,102}$=8.04, p<.001, η_p^2=.32) and Technique × Size ($F_{1.45,24.67}$=6.21, p=.012, η_p^2=.27). Regarding Technique × Environment (Figure 7h), in Flat environments, both Adaptique and StickyRay demanded significantly less rotational movement than RayCursor (both p<=.036). In contrast, in Dense and Deep environments, Adaptique and RayCursor required significantly less movement than StickyRay (all p<=.043). Additionally, in the Sparse environment, Adaptique required significantly less rotational movement than RayCursor (p=.006). For StickyRay, the Sparse environment resulted in significantly least rotational movement, followed by Deep and then Dense, with the Flat environment also requiring significantly less movement than Dense (all p<=.026). For RayCursor and Adaptique, the Sparse and Deep environments again required significantly less movement, while Dense resulted in the significantly highest movement (all p<=.01). For Technique × Size interaction (Figure 7i), Adaptique required significantly less rotational movement than StickyRay regardless of the target size, and required significantly less rotational movement than RayCursor when selecting Small targets (all p<=.023).

5.4.3 Error Rate. We defined an error as any trial with at least one missed selection or with a timeout. The error rate was determined by the number of errors divided by the total number of trials within the same condition. We included all trials in this analysis. We preprocessed the data with an Aligned Rank Transform (ART) [63] and ART-C preprocessing for post hoc analysis when relevant [16].

We found significant main effects for Technique ($F_{2,34}$=30.27, p<.001, η_p^2=.64), Size ($F_{1,17}$=16.81, p<.001, η_p^2=.50), and Environment ($F_{3,51}$=109.37, p<.001, η_p^2=.87). Post hoc analysis (Figure 7j) of Technique revealed that using Adaptique and RayCursor resulted in significantly less error rate than using StickyRay (both p<.001).

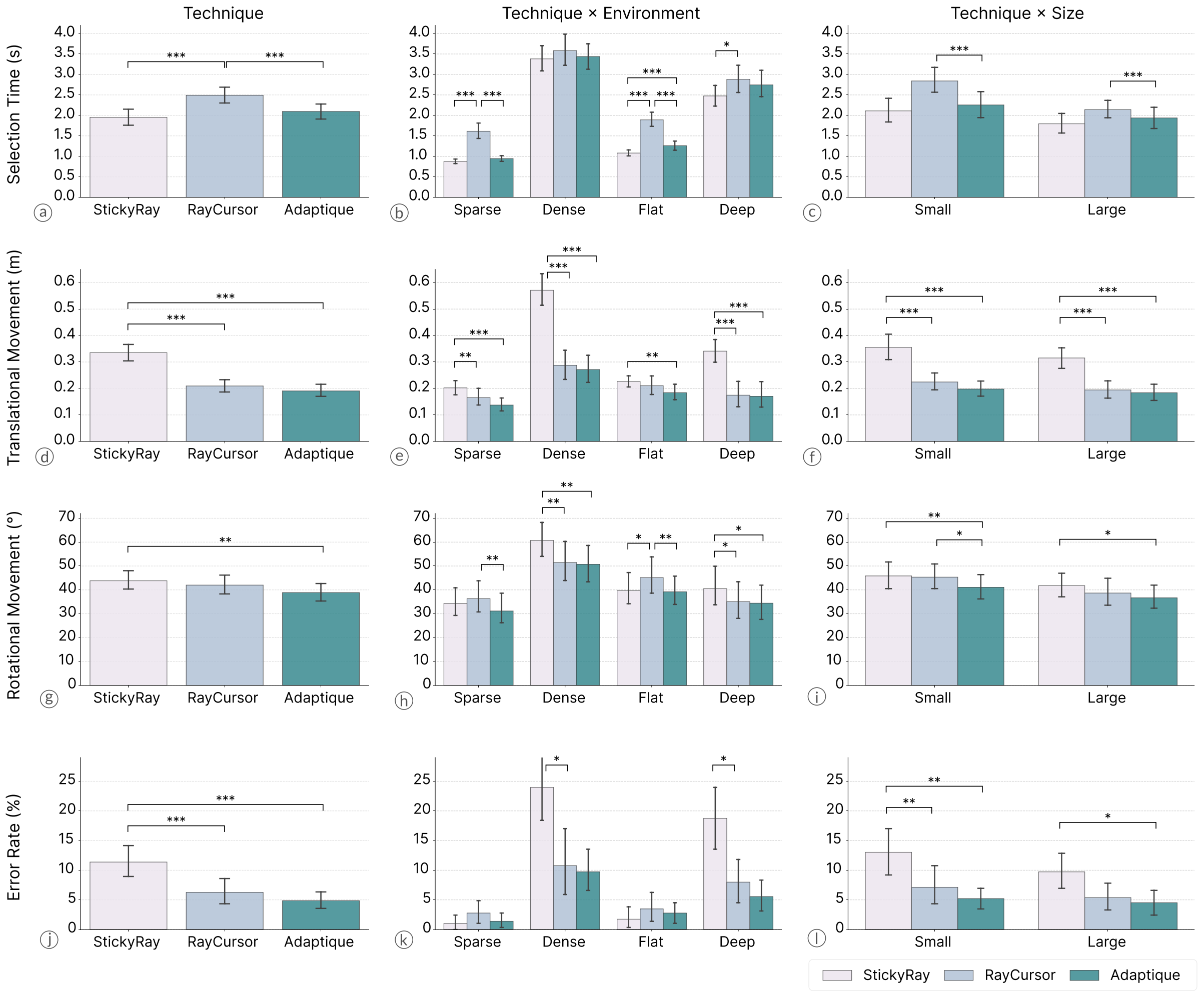

Figure 7: Mean selection time, translational movement, rotational movement, and error rate. Error bars represent the mean 95% confidence intervals. The symbol $*$ indicates $p<.05$, $**$ indicates $p\leq.01$, and $***$ indicates $p\leq.001$.

We found no significant three-way interaction for error rate. However, we observed significant two-way interactions for TECHNIQUE × ENVIRONMENT ($F_{3.51,59.72}$=13.16, p<.001, η_p^2=.44) and TECHNIQUE × SIZE ($F_{2,34}$=5.8, p=.007, η_p^2=.25). Further TECHNIQUE × ENVIRONMENT analysis (Figure 7k) revealed that in the DENSE and DEEP environment, StickyRay resulted in significantly higher error rate than RayCursor (p<=.023). While using StickyRay, selecting in a DENSE and DEEP environment results in significantly more errors than selecting in a SPARSE and FLAT environment (all p<.001). Analyzing the TECHNIQUE × SIZE interaction (Figure 7l), we found that when selecting SMALL targets, Adaptique and RayCursor resulted in significantly fewer errors than using StickyRay (all p<=.003). When selecting LARGE targets, Adaptique resulted in a significantly lower error rate than StickyRay (p=.001).

Table 1: Overview of selection technique comparison across performance metrics. ✓ indicates the techniques that were found most optimal for the given metric (as determined by statistical main effects). × indicates less performant techniques. Results showed that Adaptique was among the most optimal techniques for all metrics.

Metric	StickyRay	RayCursor	Adaptique
Selection Time	✓	×	✓
Trans. Movement	×	✓	✓
Rot. Movement	×	×	✓
Error rate	×	✓	✓

5.4.4 Summary of Quantitative Results. Our quantitative study results showed that StickyRay and RayCursor exhibit different performance advantages under different conditions and metrics. Overall, StickyRay is faster than RayCursor in all environments except for DENSE environment, while RayCursor is more precise and requires less translational movement, especially in DENSE and

Deep environments. In contrast, Adaptique consistently achieved optimal performance for all metrics in different contexts, as shown in Table 1. Although single techniques occasionally performed as well as Adaptique in specific metrics, they exhibited performance degradation in other metrics or a particular environment. For example, although StickyRay performed as well as Adaptique in selection time, it required more movement and caused more errors, especially in Dense and Deep environments. Similarly, RayCursor performed comparably to Adaptique in terms of movement and accuracy but required significantly more selection time, and even worse in Sparse and Deep environment. Adaptique performed comprehensively well in all metrics, indicating that our system effectively balanced between different objectives and technique trade-offs.

5.4.5 Questionnaire Results and Preferences. Friedman tests on usability ratings showed significant results in perceived Precision ($\chi^2(2)$=7.26, p<.05), Difficulty ($\chi^2(2)$=7.00, p<.05), and Confidence ($\chi^2(2)$=6.26, p<.05). However, Wilcoxon post hoc tests with Bonferroni correction did not show any significant differences.

The majority of the participants (ten) preferred Adaptique over the other techniques, while four chose StickyRay and four chose RayCursor. Adaptique was considered “fast”(P6), “precise” (P13), “easy to navigate” (P14), and "convenient" (P3), and combined the advantages of StickyRay and RayCursor, offering the most suitable technique for the environment (5 out of 18). P10 mentioned “both StickyRay and RayCursor are convenient in different situations. [..] So being able to switch to the other based on situations is preferred”. Meanwhile, although StickyRay was “intuitive”(P1) and “easy to use”(P12), the technique was less “precise” (P14) and “forces users to move more” (P8) in complex environments with target occlusion (expressed by 11 out of 18 participants). Regarding RayCursor, although participants liked its “full control” (P5) of cursor depth in cluttered environments, it was considered “harder” (P1) and more “tiring” (P2) due to the “additional control required” (P8), especially when there were fewer objects (9 out of 18 participants). Some participants did not prefer Adaptique because it incorporated the technique they did not like, or due to “delayed” (P5), or “distracting” (P11) switching. For example, P1 said “Adaptique is uncomfortable because it incorporated RayCursor”.

Most of the participants gave positive feedback on the switching (14 out of 18) due to its “consistency” (P8) and “accuracy” (P16) in technique selection, and “clear” (P14) feedback. Participants liked that Adaptique only switched techniques when needed and not in the middle of a selection. P8 mentioned that “the switching is pretty handy and intelligent, selecting the most efficient mode almost all the time. The mode does not vary constantly and is consistent enough for the user to get used to the selection.” Overall, visual, audio, and tactile feedback helped the participants in “notifying the technique change” (P4), and “improving the experience” (P2).

5.4.6 Technique Switching Analysis. To validate Adaptique's switching proficiency, we analyzed its switching behavior across all trials. One or more switches occurred in 53. 65% of the trials. In other trials, the current technique was considered optimal, indicating that switching occurs only when Adaptique deems it necessary. For technique switching trials, we observed that the number of switches was low, with a mean of 1.1 among the trials involving switching. These results indicate Adaptique's stability in selecting technique, as too sensitive switching may be confusing for users. Adaptique also showed strong consistency in selecting the technique according to the current context. Specifically, 99.6% of Deep environment and 100% of Dense environment trials concluded with RayCursor, while 98.3% of Flat and 96.2% of Sparse environment trials concluded with StickyRay.

In terms of timing, we found that the first switch typically occurred early in the interaction, on average at 0.37 seconds (variance = 0.08) after the trial began, compared to an average total task duration of 1.67 seconds. To further investigate the temporal relationship between switching and user movement, we compared user movement with the timing of switches. In most cases, the switch occurred before the onset of their pointing action. These results suggest that Adaptique is able to proactively switch techniques before users initiate a pointing movement, thus minimizing disruption and cognitive load. However, we also observed a small number of cases where switching occurred during or after the initial ballistic movement, possibly due to occasional delays in calculations or users pointed beyond the trial area. In these instances, users were usually able to complete the selection without initiating another pointing movement. However, in rare cases (8.5% of the trials in which a switch occurred), a second high-speed movement occurred, likely as a corrective adjustment following the delayed switch.

Overall, these findings indicate that Adaptique switches techniques efficiently, consistently, and early enough to assist users without imposing cognitive overhead or disruption to movement.

6 Discussion

We introduced Adaptique to address the selection challenges inherent in dynamic virtual environments. Adaptique adapts the selection technique according to a wide range of environments and user states based on a computational approach of extracting contextual information that effectively captures scenarios where users perceive objects as overlapping, too small, arranged differently, etc. Adaptique also considers different aspects of performance built from established selection models and balances these factors to align with the design needs.

Our results underscore the need for adaptivity, as using the same technique in various scenarios can introduce trade-offs that lead to difficulty and negatively impact performance and user experience. For example, although RayCursor is effective in precisely selecting objects, especially in Dense and Deep environments, it is generally slower due to its complexity in selecting objects, especially in Sparse and Flat environments. In contrast, Adaptique consistently achieved optimal results in terms of selection time, movement, and error rate in different contexts (Table 1). Although single techniques can perform as well as Adaptique in specific metrics, they typically exhibit performance degradation in other metrics or particular environments. These findings suggest that our method successfully identifies and applies the most suitable technique for each scenario to balance objectives and technique trade-offs.

Our application showcases Adaptique's utility and applicability in a dynamic and practical setting. Adaptique can automatically switch the selection tool to a more suitable one when the task becomes more difficult to complete with the current tool. For example, selecting book layouts on the bookshelf is easy with normal

RayCasting, while selecting books stacked on the table is difficult because they can occlude one another and require extra precision. Therefore, when users point toward the stack of books, Adaptique smoothly and proactively switches to RayCursor, ensuring smooth transitions and consistent tool usage in the new context. This adaptive behavior is driven by the interaction space spreading out from the pointing direction, which gradually captures the context of the user's attention. In addition, the thresholding mechanism ensures consistent improvement across frames before confirming a switch, preventing users from experiencing inconsistent switching.

6.1 Contextual Information Extraction

Adaptique effectively extracts fundamental contextual information as the basis for adaptation. In contrast to previous work that only extracts density information [8], we further parameterize target relationships, such as occlusion, to model objectives more accurately. This consideration is crucial because the performance depends on an object's effective size rather than its dimensions. Still, although our current approach considers occlusion relative to the controller's perspective as this directly influences performance, this may not align with how users perceive occlusion. For example, an object that appears occluded from the user's sight might not be occluded according to the controller's viewpoint. This discrepancy can create a mismatch of the optimal technique between user expectations and the model's behavior, degrading the user experience more than the performance benefits justify. Future work could explore integrating perceived contextual information to enhance user intuitiveness.

Additionally, although Adaptique is currently implemented in VR, it is intended for future extension into mixed reality environments, where physical interactable objects are rigid and the surroundings are more dynamic and unpredictable. However, there remain limitations to the tracking of the 3D positions and shapes of real-world objects. We believe that future improvements in object tracking technologies will overcome these limitations [32, 53], allowing more use cases to benefit from Adaptique.

6.2 Optimization Objectives

Adaptique integrates multiple objectives that balance performance (e.g., speed, accuracy) and usability (e.g., comfort, familiarity). This formulation, combined with contextual modeling, generalizes well to diverse XR scenarios. Building on this foundation, Adaptique can be further expanded to include more contextual information and objectives. For example, including the moving speed of targets or users [27, 38, 44] can be beneficial in scenarios such as public transportation, interactions with moving targets like people and animals, and gaming. Additionally, the contextual information could be expanded to account for factors accumulated over time, such as cumulative fatigue [17, 30] and workload [41] during prolonged interactions. More objectives can also be considered to accommodate different design requirements, such as social acceptance [15, 39, 62], engagement [51], sensor error [56], available range of motion [11, 62], and so on.

To enable adaptation in real-time, our objectives include certain assumptions and simplifications. For example, the body parameters for calculating comfort scores rely on average human data instead of individual data. While this simplification might introduce deviation from using individual parameters, the error affects all techniques equally and, therefore, has little impact on our adaptation. Alternatively, in the future users could also input their specific values for more precise estimation. In addition, we assumed users would select the target with a specific movement trajectory due to the challenges of predicting user motions. Though users might not perform the selection in this manner, this approach captures the general trends of exertion and fatigue and responsively adapts to users' posture changes. Future work could include a movement prediction model based on velocity profiles [28]. Our approach to assigning familiarity scores through pilot testing proved effective and practical in our study, but it may have limitations in scalability and subjectivity, as further pilot testing is required for different sets of techniques. Further research could develop a more systematic approach to assigning familiarity scores, such as inferring values from user preferences collected through voting on various benchmark scenes. Another direction could be training machine learning models to predict the complexity of selection techniques using behavioral data, such as cognitive load metrics [48].

6.3 Selection Techniques

Currently, we considered raycasting-based selection techniques as candidates that address various scenarios effectively. Future explorations could include more diverse pointing-based techniques, such as those with different modalities [9, 57]. This would open up a vast design space to integrate complementary techniques. For example, switching the technique to gaze when the arm is tired or occupied can be useful in prolonged use. However, modifications to the current objective score calculations would be necessary to ensure these scores remain comparable across different modalities.

To reduce confusion and required training, we limited the number of techniques to one per selection scenario. Although having more selection techniques as candidates increases the granularity of optimal choices, we suggest it could lead to excessive switchings that distract the user. While this limitation reduces distraction, future exploration could provide alternative techniques for the same selection scenario to allow users to customize their set of techniques. This flexibility would address feedback from study participants who did not prefer Adaptique as it included techniques they disliked.

Additionally, we encountered challenges in modeling advanced selection techniques. For example, RayCursor, which is controlled by a combination of ray movement and swiping on the trackpad, makes the user interaction pattern unpredictable. Thus, we simplified the model by considering only the movement of the ray and assuming the cursor remains at the correct depth. This simplification might overestimate the technique's performance as the swiping effort is not modeled, and we assume the calibration effect of the familiarity score mitigated this deviation. Future research could explore more sophisticated models for advanced techniques [52] or introduce a calibration method for the model that could not fully describe a technique. Another possible approach could be to employ data-driven methods to better understand the relationship between human behaviors and performance metrics.

6.4 Further Adaptation Considerations

Our qualitative result showed that participants were satisfied with the switching due to its consistency and accuracy in technique selection. Similarly, our quantitative results showed no significant performance degradation from potential switching or distraction side effects, as Adaptique consistently performed among the best across conditions. This outcome was achieved by optimizing system responsiveness and adjusting sensitivity through a window threshold technique to minimize delays and unnecessary switching. To further enhance switching, we could combine the current system with an intention prediction model to decide the optimal switching timing [66].

However, even though we did not identify the significant cost of switching, there might still exist subtle side effects that our current evaluation could not adequately capture. This raises the possibility that Adaptique, under optimal conditions, might outperform individual techniques, rather than merely showing comparable performance to the most effective one. To further investigate this, future studies could focus on quantifying switching costs by measuring human response time and cognitive load. These measures could also be analyzed in relation to the complexity of the techniques being switched or the degree of disparity between them [55].

7 Conclusion

We presented Adaptique, an online multi-objective model that adaptively switches to the most optimal VR selection technique based on user context and environment combined with human performance objectives. The results show that Adaptique can significantly improve selection time, movement, and error rate against the use of singular techniques. In addition, a majority of participants preferred Adaptique who expressed a positive sentiment for switching techniques when exposed to various environments. In sum, Adaptique shows that it is beneficial to switch between techniques to gain the most performance across multiple environments. Furthermore, considering multiple objectives is important to reflect the trade-offs between different techniques. Our work opens up further research on additional selection objectives, techniques, and modalities to accurately model and adapt to interactions commonly needed in our daily lives.

References

[1] Ferran Argelaguet and Carlos Andujar. 2008. Improving 3D Selection in VEs through Expanding Targets and Forced Disocclusion. In *Smart Graphics*, Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier, and Marc Christie (Eds.). Vol. 5166. Springer Berlin Heidelberg, Berlin, Heidelberg, 45–57. doi:10.1007/978-3-540-85412-8_5

[2] Ferran Argelaguet and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics* 37, 3 (May 2013), 121–136. doi:10.1016/j.cag.2012.12.003

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

[4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Software* 22, 4 (1996), 469–483. doi:10.1145/235815.235821

[5] Joanna Bergström, Tor-Salve Dalsgaard, Jason Alexander, and Kasper Hornbæk. 2021. How to Evaluate Object Selection and Manipulation in VR? Guidelines from 20 Years of Studies. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 533, 20 pages. doi:10.1145/3411764.3445193

[6] Xiaojun Bi and Shumin Zhai. 2016. Predicting Finger-Touch Accuracy Based on the Dual Gaussian Distribution Model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 313–319. doi:10.1145/2984511.2984546

[7] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. 161–168.

[8] Jeffrey Cashion, Chadwick Wingrave, and Joseph J. LaViola. 2013. Optimal 3D Selection Technique Assignment Using Real-Time Contextual Analysis. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 107–110. doi:10.1109/3DUI.2013.6550205

[9] 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* (, Christchurch, New Zealand,) *(VRST '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 11 pages. doi:10.1145/3611659.3615693

[10] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 282–297. doi:10.1145/3472749.3474750

[11] Yi Fei Cheng, Christoph Gebhardt, and Christian Holz. 2023. InteractionAdapt: Interaction-driven Workspace Adaptation for Situated Virtual Reality Environments. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, San Francisco CA USA, 1–14. doi:10.1145/3586183.3606717

[12] Choongho Chung and Sung-Hee Lee. 2024. Continuous Prediction of Pointing Targets With Motion and Eye-Tracking in Virtual Reality. *IEEE Access* 12 (2024), 5933–5946. doi:10.1109/ACCESS.2024.3350788

[13] Logan D. Clark, Aakash B. Bhagat, and Sara L. Riggs. 2020. Extending Fitts' Law in Three-Dimensional Virtual Environments with Current Low-Cost Virtual Reality Technology. *International Journal of Human-Computer Studies* 139 (July 2020), 102413. doi:10.1016/j.ijhcs.2020.102413

[14] William Delamare, Maxime Daniel, and Khalad Hasan. 2022. MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, New Orleans LA USA, 1–6. doi:10.1145/3491101.3519692

[15] Pouya Eghbali, Kaisa Väänänen, and Tero Jokela. 2019. Social Acceptability of Virtual Reality in Public Spaces: Experiential Factors and Design Recommendations. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*. ACM, Pisa Italy, 1–11. doi:10.1145/3365610.3365647

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

[17] João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. 2021. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–11. doi:10.1145/3411764.3445349

[18] João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. 2022. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–16. doi:10.1145/3526113.3545651

[19] Paul M. Fitts. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391. doi:10.1037/h0055392

[20] Andris Freivalds. 2011. *Biomechanics of the Upper Limbs: Mechanics, Modeling and Musculoskeletal Injuries*. CRC press.

[21] Jenny Gabel, Susanne Schmidt, Oscar Ariza, and Frank Steinicke. 2023. Redirecting Rays: Evaluation of Assistive Raycasting Techniques in Virtual Reality. In *29th ACM Symposium on Virtual Reality Software and Technology*. ACM, Christchurch New Zealand, 1–11. doi:10.1145/3611659.3615716

[22] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New Orleans LA USA, 197–208. doi:10.1145/3332165.3347933

[23] Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Portland Oregon USA, 281–290. doi:10.1145/1054972.1055012

[24] Tovi Grossman and Ravin Balakrishnan. 2006. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. ACM, Montreux Switzerland, 3–12. doi:10.1145/1166253.1166257

[25] Tovi Grossman, Nicholas Kong, and Ravin Balakrishnan. 2007. Modeling Pointing at Targets of Arbitrary Shapes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 463–472. doi:10.1145/1240624.1240700

[26] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (June 2017), 1706–1724. doi:10.1109/TVCG.2016.2543720

[27] Khalad Hasan, Tovi Grossman, and Pourang Irani. 2011. Comet and Target Ghost: Techniques for Selecting Moving Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vancouver BC Canada, 839–848. doi:10.1145/1978942.1979065

[28] Rorik Henrikson, Tovi Grossman, Sean Trowbridge, Daniel Wigdor, and Hrvoje Benko. 2020. Head-Coupled Kinematic Template Matching: A Prediction Model for Ray Pointing in VR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14. doi:10.1145/3313831.3376489

[29] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1063–1072. doi:10.1145/2556288.2557130

[30] Christoph Albert Johns, João Marcelo Evangelista Belo, Clemens Nylandsted Klokmose, and Ken Pfeuffer. 2023. Pareto Optimal Layouts for Adaptive Mixed Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–7. doi:10.1145/3544549.3585732

[31] Angus Johnson. 2024. *Clipper2: A Polygon Clipping and Offsetting Library*. https://www.angusj.com/clipper2/Docs/Overview.htm

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. arXiv:2304.02643 [cs]

[33] Regis Kopper, Felipe Bacim, and Doug A. Bowman. 2011. Rapid and Accurate 3D Selection by Progressive Refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 67–74. doi:10.1109/3DUI.2011.5759219

[34] 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

[35] Jérémy Lacoche, Thierry Duval, Bruno Arnaldi, Eric Maisel, and Jérôme Royan. 2019. Machine Learning Based Interaction Technique Selection for 3D User Interfaces. In *Virtual Reality and Augmented Reality*, Patrick Bourdot, Victoria Interrante, Luciana Nedel, Nadia Magnenat-Thalmann, and Gabriel Zachmann (Eds.). Vol. 11883. Springer International Publishing, Cham, 33–51. doi:10.1007/978-3-030-31908-3_3

[36] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Poupyrev. 2017. *3D User Interfaces: Theory and Practice* (second edition ed.). Addison-Wesley, Boston.

[37] Joong-Jae Lee and Jung-Min Park. 2020. 3D Mirrored Object Selection for Occluded Objects in Virtual Environments. *IEEE Access* 8 (2020), 200259–200274. doi:10.1109/ACCESS.2020.3035376

[38] Nianlong Li, Feng Tian, Jin Huang, Xiangmin Fan, and Hongan Wang. 2018. 2D-BayesPointer: An Implicit Moving Target Selection Technique Enabled by Human Performance Modeling. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–6. doi:10.1145/3170427.3188520

[39] Zhipeng Li, Christoph Gebhardt, Yves Inglin, Nicolas Steck, Paul Streli, and Christian Holz. 2024. SituationAdapt: Contextual UI Optimization in Mixed Reality with Situation Awareness via LLM Reasoning. 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 43, 13 pages. doi:10.1145/3654777.3676470

[40] David Lindlbauer. 2022. The Future of Mixed Reality Is Adaptive. *XRDS: Crossroads, The ACM Magazine for Students* 29, 1 (Sept. 2022), 26–31. doi:10.1145/3558191

[41] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New Orleans LA USA, 147–160. doi:10.1145/3332165.3347945

[42] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 35–43. doi:10.1109/VR46266.2020.00021

[43] I. Scott MacKenzie and William Buxton. 1992. Extending Fitts' Law to Two-Dimensional Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '92*. ACM Press, Monterey, California, United States, 219–226. doi:10.1145/142750.142794

[44] 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 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

[45] Lynn McAtamney and E. Nigel Corlett. 1993. RULA: A Survey Method for the Investigation of Work-Related Upper Limb Disorders. *Applied Ergonomics* 24, 2 (April 1993), 91–99. doi:10.1016/0003-6870(93)90080-S

[46] Mark R Mine. 1995. *Virtual environment interaction techniques*. Technical Report. UNC Chapel Hill CS Dept.

[47] Elena Molina and Pere-Pau Vázquez. 2023. Two-Step Techniques for Accurate Selection of Small Elements in VR Environments. *Graphical Models* 128 (July 2023), 101183. doi:10.1016/j.gmod.2023.101183

[48] Johanna Renny Octavia, Karin Coninx, and Kris Luyten. 2011. Squeeze Me and i'll Change: An Exploration of Frustration-Triggered Adaptation for Multimodal Interaction. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 79–86. doi:10.1109/3DUI.2011.5759221

[49] Johanna Renny Octavia, Chris Raymaekers, and Karin Coninx. 2011. Adaptation in Virtual Environments: Conceptual Framework and User Models. *Multimedia Tools and Applications* 54, 1 (Aug. 2011), 121–142. doi:10.1007/s11042-010-0525-z

[50] Antti Oulasvirta, Niraj Ramesh Dayama, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. 2020. Combinatorial Optimization of Graphical User Interface Designs. *Proc. IEEE* 108, 3 (March 2020), 434–464. doi:10.1109/JPROC.2020.2969687

[51] Henning Pohl and Roderick Murray-Smith. 2013. Focused and Casual Interactions: Allowing Users to Vary Their Level of Engagement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2223–2232. doi:10.1145/2470654.2481307

[52] Adrian Ramcharitar and Robert J. Teather. 2017. A Fitts' Law Evaluation of Video Game Controllers: Thumbstick, Touchpad and Gyrosensor. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 2860–2866. doi:10.1145/3027063.3053213

[53] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. SAM 2: Segment Anything in Images and Videos. arXiv:2408.00714 [cs.CV] https://arxiv.org/abs/2408.00714

[54] David Saffo, Sara Di Bartolomeo, Caglar Yildirim, and Cody Dunne. 2021. Remote and Collaborative Virtual Reality Experiments via Social VR Platforms. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–15. doi:10.1145/3411764.3445426

[55] Joey Scarr, Andy Cockburn, Carl Gutwin, and Philip Quinn. 2011. Dips and Ceilings: Understanding and Supporting Transitions to Expertise in User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vancouver BC Canada, 2741–2750. doi:10.1145/1978942.1979348

[56] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. 2020. Outline Pursuits: Gaze-assisted Selection of Occluded Objects in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–13. doi:10.1145/3313831.3376438

[57] 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 (Nov. 2022), 3585–3595. doi:10.1109/TVCG.2022.3203096

[58] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2006. Object Selection in Virtual Environments with an Improved Virtual Pointer Metaphor. In *Computer Vision and Graphics: International Conference, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings*, K. Wojciechowski, B. Smolka, H. Palus, R.S. Kozera, W. Skarbek, and L. Noakes (Eds.). Springer Netherlands, Dordrecht, 320–326. doi:10.1007/1-4020-4179-9_46

[59] Tomu Tahara, Takashi Seno, Gaku Narita, and Tomoya Ishikawa. 2020. Retargetable AR: Context-aware Augmented Reality in Indoor Scenes Based on 3D Scene Graph. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, Recife, Brazil, 249–255. doi:10.1109/ISMAR-Adjunct51615.2020.00072

[60] The CGAL Project. 2024. *CGAL User and Reference Manual* (5.6.1 ed.). CGAL Editorial Board. https://doc.cgal.org/5.6.1/Manual/packages.html

[61] Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*. doi:10.1109/3DUI.

2007.340783

[62] Julie R. Williamson, Mark McGill, and Khari Outram. 2019. PlaneVR: Social Acceptability of Virtual Reality for Aeroplane Passengers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. doi:10.1145/3290605.3300310

[63] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 143–146. doi:10.1145/1978942.1978963

[64] Huiyue Wu, Xiaoxuan Sun, Huawei Tu, and Xiaolong Zhang. 2023. ClockRay: A Wrist-Rotation Based Technique for Occluded-Target Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–12. doi:10.1109/TVCG.2023.3239951

[65] Enes Yigitbas, Joshua Heindörfer, and Gregor Engels. 2019. A Context-aware Virtual Reality First Aid Training Application. In *Proceedings of Mensch Und Computer 2019*. ACM, Hamburg Germany, 885–888. doi:10.1145/3340764.3349525

[66] Difeng Yu, Ruta Desai, Ting Zhang, Hrvoje Benko, Tanya R. Jonker, and Aakar Gupta. 2022. Optimizing the Timing of Intelligent Suggestion in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–20. doi:10.1145/3526113.3545632

[67] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments. *ACM Transactions on Graphics* 38, 6 (Dec. 2019), 1–13. doi:10.1145/3355089.3356544

[68] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Fully-Occluded Target Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (Feb. 2020), 3402–3413. doi:10.1109/TVCG.2020.3023606