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

Anonymous Author(s)

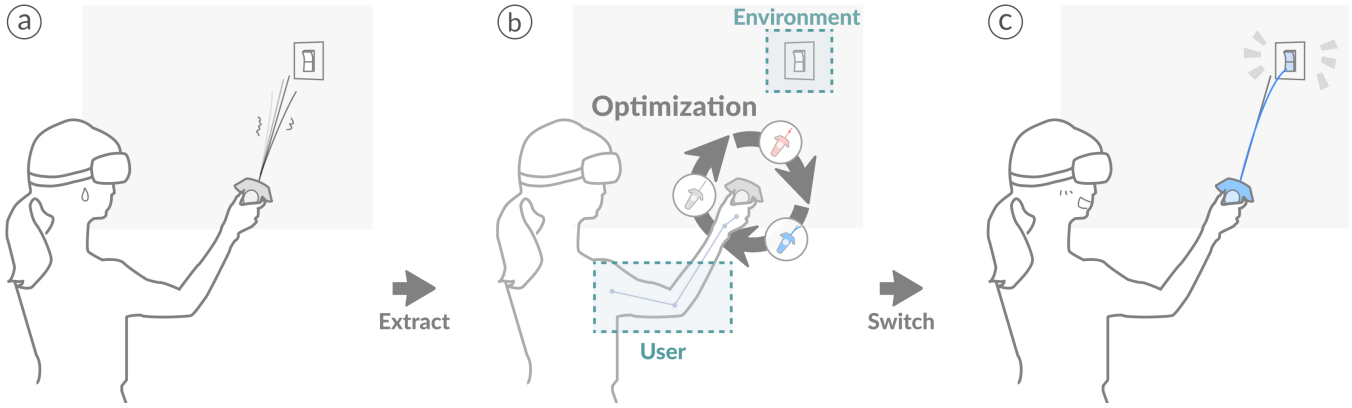

Figure 1: Adaptique switches the selection technique based on environmental and user-based factors, and considers multiple objectives for VR selection. In this example, the user attempts to select the light switch on the far wall to light up the room. (a) Since the light switch is small and far, the 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.

CHI '25, April 26-May 1, 2025, Yokohama, Japan

ACM ISBN xxx…$15.00
https://doi.org/xxx

CCS CONCEPTS

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

KEYWORDS

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

ACM Reference Format:
Anonymous Author(s). 2025. Adaptique: Multi-objective and Context-aware Online Adaptation of Selection Techniques in Virtual Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/xxx

1 INTRODUCTION

Selection tasks in extended reality (XR) can be challenging in dynamic environments due to factors such as small and 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 [38]. For example, a user might be selecting buttons on a large panel, which requires only a simple and fast 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 tool designed to target fine details in a dense environment. Following that, they might interact with an IoT light switch on a distant wall to view a real-world object clearly. Because the switch is located far away from their reach, they need a tool

that can handle distant objects effectively. These scenarios represent three distinct environments, making it difficult to use a single selection tool for all tasks.

Previous work has addressed some of the selection challenges in XR and 3D user interfaces, such as selecting small, distant, occluded objects or selecting in a dense environment [55, 65]. However, these techniques are often tailored to specific scenarios and become overly complicated when applied outside of 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.

In this research, we propose Adaptique, a virtual reality (VR) system that adaptively selects the most suitable selection technique (Figure 1). Inspired by previous work on context-aware adaptation of interfaces in the layout of virtual contents [9, 39, 47], we frame our problem as a multi-objective online optimization problem. We based our adaptation on environmental factors such as the object's position, size, and relationship to other objects, as well as user-based factors such as the user's current posture and familiarity with the techniques (Figure 1b). We identified four objectives, including Speed, Accuracy, Comfort, and Familiarity. We formalized these objectives with established performance metrics that could transform those factors into our criteria. 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 normal RayCasting, StickyRay [22, 40], and RayCursor [3] to effectively switch between normal selection, small targets, dense environments, and target occlusion. We demonstrated Adaptique's utility and applicability in an 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 VR 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 technique in XR.
- An application with various selection tasks that showcases the versatility and utility of Adaptique.
- The results of a user study demonstrate the benefits of using Adaptique in various environments.

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 [35, 44]. 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 [23, 32, 33, 45], snapping mechanisms to decrease the precision requirement by enlarging the effective size [20, 40, 55]. 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 [23], RayCursor [3], ClockRay [61], and Alpha Cursor [65] 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 [13] 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 [36].

Although these techniques make the selection task easier in their designed cases, they are usually too complicated and unnecessary when applied outside of their intended context. In contrast, Adaptique adapts the interaction technique to the current context, making selection easier in any scenario that users might encounter 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 in virtual environment, focusing on speed, accuracy, and comfort. We evaluate these factors using Fitts' Law [18] for speed, the end-point distribution model [64] for accuracy, and the Consumed Endurance model [28] for comfort.

In Fitts' law, the predicted time needed to select a target based on the target's distance and width is formulated as $MT = a + b \cdot log_2(\frac{2A}{W})$ [18]. 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 in tasks of higher dimension space. 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 [41]. We adopted this model due to its simplicity, its ability to deal with non-rectangle 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 [20] 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 [64]. As for the interaction that requires a higher degree of freedom in translation, such as virtual hand pointing, the 3D Fitts' Law is used more frequently [12, 51].

The endpoint distribution model describes 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 [64]. Combined with Bi's method [6], this model can also estimate the 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) [28] and RULA [43] 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 to the context in terms of 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 [9, 10, 16, 17, 21, 25, 29, 39, 56], 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 [9, 56], ergonomics [16, 29], physical space [10], or user's intention [21]. 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 virtual content layout where elements can be freely moved and placed. In contrast, our work assumes that the content and interactable targets in XR are relatively static due to being physical objects that users may want to keep intact, and to minimize changes in the environments. Instead, we believe that the user and their interaction should adapt to their current context. Inspired by those layout adaptation works, we employed a multi-objective optimization framework for adaptation due to its simplicity, scalability, and ability to balance multiple objectives in a controllable and clear way [47].

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 a virtual environment [7, 34, 46], these studies often remain conceptual, do not comprehensively address dynamic environmental factors and user states in real-time, focus more on personalization, or on singular objectives that do not reflect the different trade-offs between selection techniques. Other works have changed the modality of selection techniques based on its availability or stability. For example, Sidenmark et al. [54] switched the gaze input to the controller or head when the quality of the gaze signal dropped, and Yigitbas et al. [62] switched to the gaze control when controllers were unavailable. Recent work has included selection techniques in the XR layout adaptation [10]. In contrast, our work focuses on a real-time adaptation system for the interaction technique itself based on environmental and user-based contextual information and considers multiple objectives for interaction.

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 selection task performance 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 higher levels of abstraction. Overcoming these drawbacks necessitates user familiarity [52]. 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. Technically, the system works in the following steps as illustrated in the pipeline of Figure 2:

- Acquire the targets within the interaction space.
- Extract contextual information, including user postures, target positions, sizes, and so on.
- Calculate and aggregate the objectives for each technique.
- Switch the technique if a more optimal one shows a consistent improvement in overall performance.

3.1 Extracting Contextual Information

We first extract contextual information from the environment and the user that will be used as input for adaptation. We define the interaction space as a cone with a radius of r_c originating from the pointing direction. This space represents the user's temporal area of attention, ensuring that only targets that are relevant for interaction are considered for adaptation, and also reduces the required computation. Every object within this area is included as input for adaptation.

After defining the set of objects for interaction, we provide positions and sizes for all targets in 3D and 2D space relative to the controller. We provide target information in both 3D and 2D space as some techniques (i.e. common RayCasting) do not consider depth-based information and effective target sizes are therefore affected by occlusion, while some techniques (i.e. RayCursor [3] and GazeRayCursor [8]) utilize depth-based information of targets which ignores occlusion. For 3D information, we simply provide the 3D target positions, shapes, and sizes relative to the controller position. For the targets' 2D space, we project all targets' 3D meshes onto a plane perpendicular to the controller's pointing direction. The projected targets are scaled to ensure that their visual size remains consistent. To incorporate occlusion, we calculate a convex mesh of

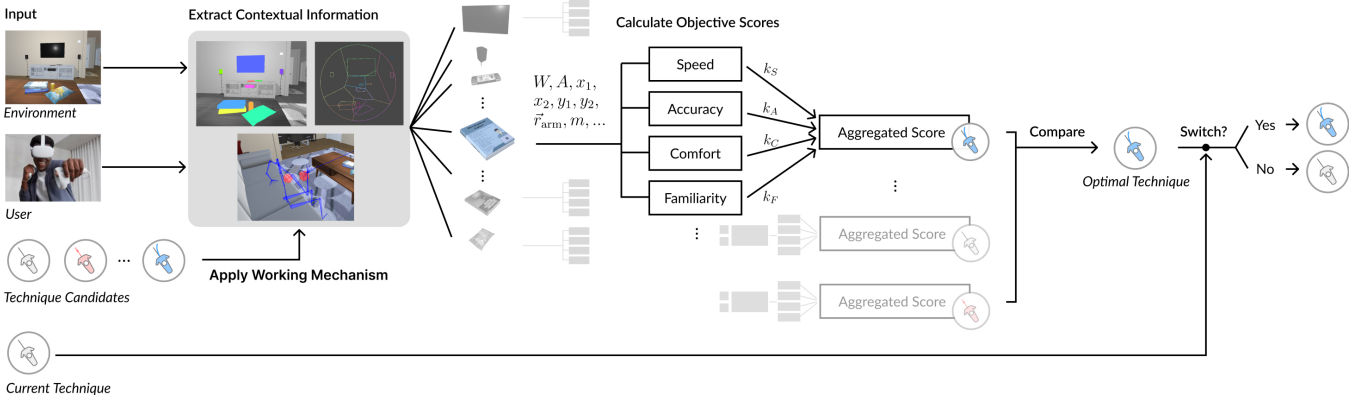

Figure 2: The Adaptique pipeline extracts user and environmental input, applies the working mechanism of selection techniques to calculate model input for each selectable object and techique, aggregates the objects' objective scores for each technique, finds the optimal technique for the interactable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

the projected vertices to form an outline polygon using CGAL [57]. The outline polygon is then clipped by other outline polygons that occlude the object through Clipper2 [30]. Finally, we recalculate the target centroid (c) using the final clipped polygon. For each target, we provide the final 2D outline polygons that define their activation region along with their position relative to the controller position.

In addition, we provide information about the posture of the user and their current selection technique as input. As most VR systems only provide tracking via controllers and the head-mounted display (HMD), we generate and provide the user's current posture based on inverse kinematics using the HMD and controller position as input. The 2D and 3D environmental information is updated every frame along with the user information.

3.2 Objectives

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. For our implementation of Adaptique, we use common performance metrics and formalize them using established models of human performance and movement. The system can easily 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 based on the widely used index of difficulty (ID) formula in the Shannon formulation based on Fitts' Law

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

which states that the difficulty of selection, and thereby the speed of selection, increases with larger target amplitude and smaller target width [41]. The formula relies on two main parameters, movement amplitude A, and the target size W, both visual angular (Figure 3a). For this work, we define W as the effective width of the target activation region along the pointing path, defined as a vector between the current pointing direction and the target centroid, which has been shown to play a greater role than its visual boundaries on selection time [24, 58]. The movement amplitude A is the angular distance the ray needs to travel along that path to the aiming center of the target. Here, we use the centroid point of the activation region as the aiming center. To have a higher score representing higher performance, 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. However, since all techniques in this paper's 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. We utilize the *EDModel* by Yu et al. [64] to formalize our accuracy objective (S_A). 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 from regression using collected pointing endpoint selection data [64]. 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 integral calculation, we approximate D using a rectangle defined by coordinates (x_1, x_2, y_1, y_2), where x_1, x_2, y_1 and y_2 are the intersection points on the boundary of the activation region with the axes (Figure 3b). Equation 2 can then be simplified to

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

3.2.3 Comfort. To define comfort (S_C), we use a modified Strength metric from the Consumed Endurance (CE) model to quantify arm fatigue caused by selecting a target [28]. In the CE model, the torque exerted on the shoulder muscle must match the gravity torque $\vec{g}$:

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

Where $\vec{r}_{\text{arm}}$ is the distance from the shoulder joint to the center of mass of the arm, and m is the mass of the arm. As users need to move their arms to reach a target, we calculate the shoulder torque for the predicted poses that users will perform during pointing. For simplicity, we assume that users will move the ray toward the aiming center of the target (defined in subsubsection 3.2.1) along the direction of movement by rotating their forearm with a fixed elbow position and positioning their wrist in a way that the ray is in the same direction as the forearm.

We also assume that users move their forearms at a constant rotational speed. Furthermore, we know that longer interaction time requires more energy and can lead to fatigue when endurance limits are reached [28]. To quantify this effect of time in energy, we sum up the torques of all postures (derived from the initial user posture described in subsection 3.1) along the pointing movement path, generated at rotational increments of β degrees toward the final rotation (Figure 3c). This sum represents the exertion required for a selection, where a higher exertion represents lower comfort. To keep consistency that a higher score represents less exertion, we negate the exertion:

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

3.2.4 Familiarity. Although a more advanced technique may be more efficient as defined by our performance objectives, users may still want to use simple techniques when they are sufficient to reduce effort and cognitive load. As such, each technique is assigned a "Familiarity" score

$$S_F = S_{F,\,tech} \text{ if Technique} = tech, \quad (6)$$

which represents the user's familiarity with the technique and the technique's complexity. Therefore, simpler techniques are expected to have higher familiarity scores than advanced techniques that require more interaction steps. In our implementation, we determine the familiarity for each technique based on pilot testing for simple selection tasks. In the future, we envision the potential for individual adaptation based on user exposure and performance with different interaction techniques [52]. However, this remains a subject for future investigation.

3.2.5 Normalization and Aggregation. After computing the objective scores for each selection technique and all objects within the interaction space, the objective scores are aggregated and normalized (between 0 and 1) to arrive at a single value that represents the overall score for a specific technique. We use Min-Max normalization, where the minimum (s_{min}) and maximum (s_{max}) represent

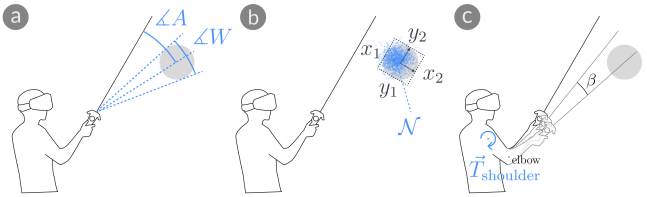

Figure 3: Scoring parameters used for objectives (a) Speed, (b) Accuracy, and (c) Comfort. The grey sphere is the target.

the theoretical limits of each objective model. For limits that do not have a theoretical bound, we consider extreme cases in our implementation. 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.

In addition, 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

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

For aggregation, we calculate the average of each objective scores 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 [11, 27] could be deployed to give objects that are more likely to be interacted with a higher priority.

3.3 Technique Switching

The most optimal interaction technique will then be decided by considering the comprehensive result of the objectives

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

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 weighting 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 in switching, 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 technique switching. To aid users in noticing technique switches, we applied haptic and audio feedback. The controller and ray also change to a unique color for visual feedback.

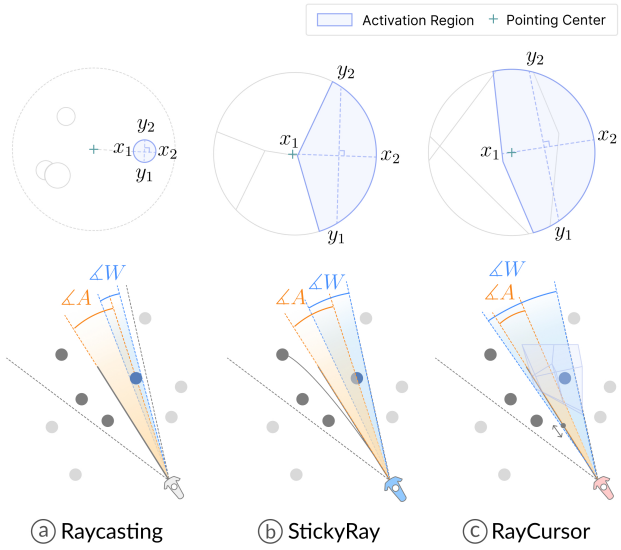

Figure 4: Context score implemented for 3 techniques (a) RayCasting, (b) StickyRay, and (c) RayCursor. The lower figure visualizes the effective size (W) and amplitude (A) of the blue target. The upper figure shows the projection of the activation region of the target.

3.4 Selection Techniques

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. We intentionally omitted techniques that require extra sensing, hardware or modalities beyond a typical VR controller. However, note that Adaptique can easily be expanded to include more 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, we introduce each included technique and how we define these variables to calculate our objectives.

3.4.1 RayCasting. RayCasting (Figure 4a) represents the most basic pointing technique in which the user points with a ray that originates 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 angle from the current pointing direction to the center of the object in the 2D space as defined in subsection 3.1. W is then the angular width of the object along the pointing direction. x_1 and x_2 are then defined as the entry and exit points, respectively, of the 2D object along the pointing direction, while y_1 and y_2 are defined as the outline points along a line that is perpendicular to the pointing direction and crosses the center of the object. For completely occluded targets, we assign the minimum score to each objective for that target. Notes that users might change the point of view to make targets unoccluded. In this case the system will reflect the change as it continuously adjusts in real-time as users complete the selection. Despite RayCasting being the most widely used technique, users' inherent hand tremors can result in instability in pointing accuracy, 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 selection of small targets. StickyRay is based on the Bubble Cursor metaphor [22], 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 selection of small targets easier. To show the current closest object, a second ray bends toward the closest object [55]. In our implementation, 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 [40]. 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 process is equivalent to finding the Voronoi region in projected 2D space, with each point's Euclidean distances defined by their angular distance to the ray direction. 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 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 mechansism is that a target will always be highlighted for selection, which may not always be preferable depending on the context of use. Finally, although StickyRay is proficient in selecting small targets and selection in sparse environments, the benefit of the technique diminishes in crowded environments and when occlusion is present.

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 choose 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 a selection with RayCursor can be performed solely with moving the ray or a combination of moving the ray and swiping on the trackpad to move the cursor, 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 (Figure 4c). 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.

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. [64] 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. [28] to calculate the center of mass for r and m. As input, we used the following values specified by Freivalds [19]: 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 the versatility and benefits of Adaptique in general selection tasks, we developed an everyday indoor VR environment where users interact with IOTs, books, food, and UI elements. 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 $r_c = 20°$, 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 $\beta = 1°$ to derive interaction postures.

To exemplify the benefits of Adaptique, we detail a walkthrough of the application. 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 now near and big and therefore easy to select (Figure 5d).

The kitchen is cluttered with many foods, ingredients, 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 have, 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 to be easy to interact with, 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 setting of the 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 evaluate the effects of Adaptique and technique switching on selection performance. This was done in the context of a controlled task where participants had to select one target among many in different environments. We compared Adaptique to using only StickyRay and RayCursor. As the study focused mainly on performance metrics, we tested a simplified version that adapts between two techniques that are known to have complementary advantages to validate the concept of Adaptique. As such, we did not include RayCasting as a baseline nor as a technique used in Adaptique.

5.1 Task

Participants were tasked to select a target object amongst many distraction targets. We varied the size of the selectable target, and

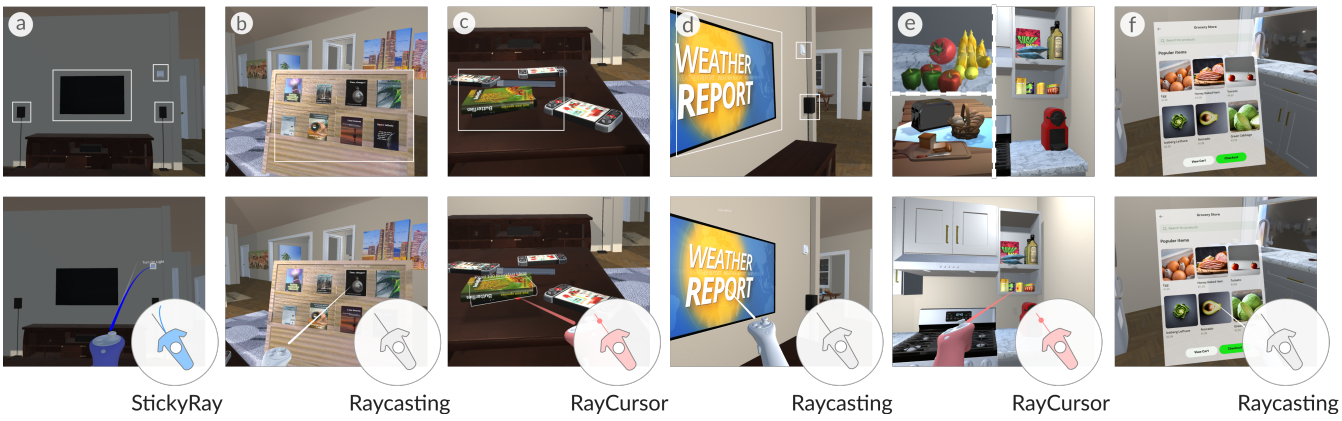

Figure 5: Adaptique chooses different techniques for different scenarios in the application, such as interacting with (a) IOTs on the far wall, (b) books layout on the nearby bookshelf, (c) books stacked on the coffee table, (d) IOTs on the side, (e) ingredients in the cluttered kitchen, and (f) UI panel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

the number of targets and density of the environment. We considered two size conditions: large (2.5°) and small (0.5°), and four different environments that varied in distractor amount and density to cover both simple and extreme cases that users might encounter in XR. The environments were also balanced, so that two environment types exploited the advantages of StickyRay, and similarly two environments exploited the advantages of RayCursor. To start the task, the user first had to align the pointer to a central position. After alignment, the target object and the distractors were displayed. The participants had to select the target object as fast and accurately as possible. The distractors were primitive shapes (cubes, spheres, cylinders, and capsules), in pseudo-random positions and sizes (2-4°), and random rotations. The distractor targets were semitransparent to minimize the effect of visual search in dense and occluded settings. No objects intersected with each other. 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 environments were as follows:

- **Sparse:** In the 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 the 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 the 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 the Flat environment, the arrangement of objects extended more in the depth direction.

We pregenerated 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

5.2 Apparatus and Participants

The interaction techniques and Adaptique were implemented as described in section 3 except that RayCasting was not included as a baseline or in Adaptique. 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: $F_{\text{StickyRay}} = 0.7$, $F_{\text{RayCursor}} = 0.3$. The rest of the parameters are the same as those in section 4.

We recruited 18 participants to carry out 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.

5.3 Procedure

Upon arrival, participants completed a consent form and a demographic questionnaire before being briefed about the study. The participants would then be placed in the correct standing position and put on the HMD. The participants then performed a practice session to familiarize themselves with each selection technique. Participants then performed all selections using each technique. In each technique section, and 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. Participants needed to select a central "ready"

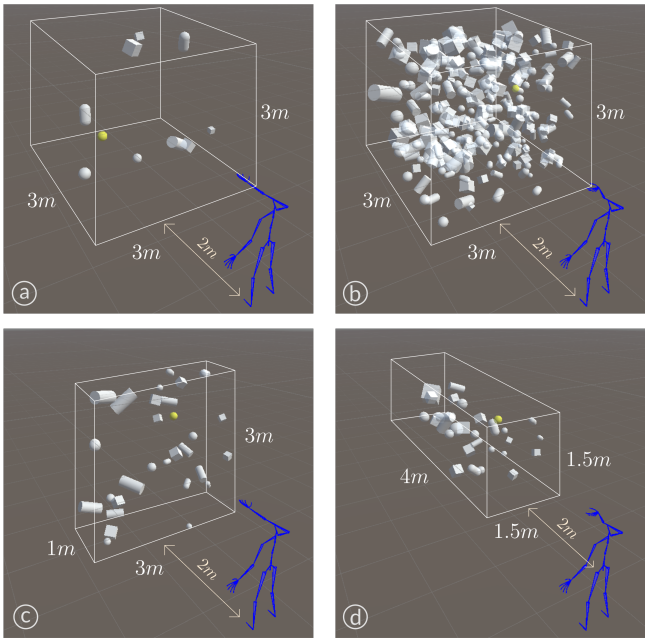

Figure 6: Environments used in the study, including (a) Sparse (b) Dense (c) Flat, and (d) Deep.

panel before starting the trial. This served as a rest period and ensured that users started the next trial from a central position. The participant then selected the target object that was highlighted in yellow. 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 for subjective feedback to capture their experiences and rested before moving on to the next technique. Participants were asked to make practice selections with the current technique to refresh their memory before starting. The study was concluded with a questionnaire for preferences and feedback. In total, we collected 18 participants × 3 Technique × 2 Target size × 4 Target environment × 8 repetitions = 3456 selections.

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 the normality of the data group 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). Questionnaire scores were analyzed using Friedman tests, and Bonferroni-corrected Wilcoxon signed-rank tests were used 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

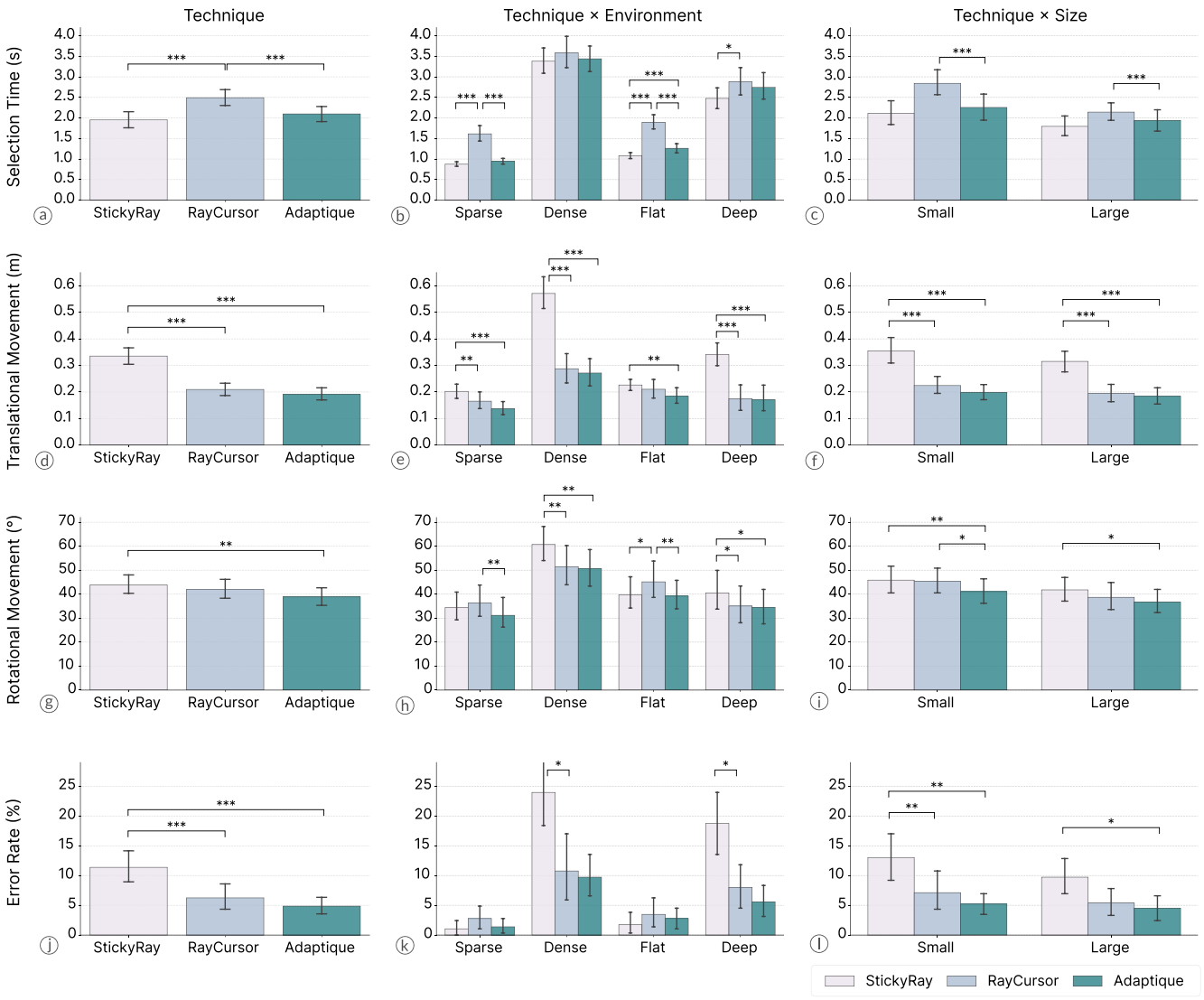

Figure 7: Mean selection time, translational movement, rotational movement, and error rate. Error bars represent the mean 95% confidence intervals. The symbol $*$ indicates $p<.05$, $**$ indicates $p\leq.01$, and $***$ indicates $p\leq.001$.

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) [60] and ART-C preprocessing for post hoc analysis when relevant [15].

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).

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).

5.4.4 Summary of Quantitative Results. Our quantitative study results showed that Adaptique consistently achieved optimal performance for all metrics in different contexts. 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 environments. In contrast, 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 technique 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).

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 study results underscore the need for adaptivity, as using the same technique in various scenarios can lead to difficulty and negatively impact performance and user experience. Specifically, our performance metric results showed that Adaptique consistently achieved optimal results in terms of selection time, movement, and error rate in different contexts. Although single techniques can perform as well as Adaptique in specific metrics, they typically exhibit performance degradation in other metrics or in particular environments. This shows the benefit of including multiple objectives in our system to reflect the trade-offs between techniques. Furthermore, we envision future work where more objectives can be considered to accommodate different design requirements, such as social acceptance [14, 59], engagement [48], sensor error [53], available range of motion [59], and more.

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 towards the stack of books, Adaptique smoothly and proactively switch 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.

During the development of Adaptique, we found that timing and sensitivity of the switch are essential for the user experience. In early exploration and pilot tests, users were unsatisfied with the adaptation due to distracting and unexpected switching. This experience was because the system was not responsive enough, limited by the complexity of geometry computation that caused unacceptable delays. Optimizing the system to enhance its responsiveness and adjusting the sensitivity using our window threshold to avoid excessive switching significantly improved the user experience. An interesting future direction would be to combine the current system with a user intention prediction model to decide the optimal switching time [63]. We can also apply predictive models to prioritize objects with which the user is more likely to interact [27], or dynamic weighting models that can adjust objective weightings as user context changes [29].

We considered a specific set of environmental scenarios and selection techniques as input that we believe shows the generalizability of Adaptique for selection in common scenarios of usage. However, Adaptique can easily be expanded to include more contextual information and techniques. For example, including the moving speed of targets or users as input for the selection of moving targets [26, 37, 42] can be beneficial in scenarios such as public transportation, interactions with moving targets like people and animals, and gaming. The contextual information could also be expanded to consider the history interactions to include factors such as fatigue [16, 29] and workload [39] accumulated over time. In addition, having more techniques, including those with different modalities [8, 54], opens 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. Adaptique could also provide alternative techniques for the same selection scenario to allow users to customize their set of techniques. This would address issues raised by study participants who did not prefer Adaptique as it included techniques they disliked.

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 positions and shapes of real-world objects. We believe that future improvements in object tracking technologies will overcome these limitations [31, 50], allowing more use cases to benefit from Adaptique. Another limitation we encountered during implementation relates to modeling advanced interaction 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 decided to simplify the model by considering only the movement of the ray and assuming the cursor remains at the correct depth. Future research could explore more sophisticated models for RayCursor and other advanced techniques [49]. A possible approach can also be to employ data-driven methods to establish the relationship between contextual information and performance metrics.

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. https://doi.org/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. https://doi.org/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. https://doi.org/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. https://doi.org/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. https://doi.org/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. https://doi.org/10.1145/2984511.2984546

[7] 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. https://doi.org/10.1109/3DUI.2013.6550205

[8] 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. https://doi.org/10.1145/3611659.3615693

[9] 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. https://doi.org/10.1145/3472749.3474750

[10] 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. https://doi.org/10.1145/3586183.3606717

[11] 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. https://doi.org/10.1109/ACCESS.2024.3350788

[12] 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. https://doi.org/10.1016/j.ijhcs.2020.102413

[13] 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. https://doi.org/10.1145/3491101.3519692
[14] 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. https://doi.org/10.1145/3365610.3365647
[15] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. 2021. An Aligned Rank Transform Procedure for Multifactor Contrast Tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 754–768. https://doi.org/10.1145/3472749.3474784
[16] 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. https://doi.org/10.1145/3411764.3445349
[17] 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. https://doi.org/10.1145/3526113.3545651
[18] 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. https://doi.org/10.1037/h0055392
[19] Andris Freivalds. 2011. *Biomechanics of the Upper Limbs: Mechanics, Modeling and Musculoskeletal Injuries*. CRC press.
[20] 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. https://doi.org/10.1145/3611659.3615716
[21] 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. https://doi.org/10.1145/3332165.3347933
[22] 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. https://doi.org/10.1145/1054972.1055012
[23] 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. https://doi.org/10.1145/1166253.1166257
[24] 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. https://doi.org/10.1145/1240624.1240700
[25] 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. https://doi.org/10.1109/TVCG.2016.2543720
[26] 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. https://doi.org/10.1145/1978942.1979065
[27] 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. https://doi.org/10.1145/3313831.3376489
[28] 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. https://doi.org/10.1145/2556288.2557130
[29] 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. https://doi.org/10.1145/3544549.3585732
[30] Angus Johnson. 2024. *Clipper2: A Polygon Clipping and Offsetting Library*. https://www.angusj.com/clipper2/Docs/Overview.htm
[31] 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]
[32] 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. https://doi.org/10.1109/3DUI.2011.5759219
[33] 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. https://doi.org/10.1145/3173574.3173655
[34] 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. https://doi.org/10.1007/978-3-030-31908-3_3
[35] 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.
[36] Joong-Jae Lee and Jung-Min Park. 2020. 3D Mirrored Object Selection for Occluded Objects in Virtual Environments. *IEEE Access* 8 (2020), 200259–200274. https://doi.org/10.1109/ACCESS.2020.3035376
[37] 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. https://doi.org/10.1145/3170427.3188520
[38] David Lindlbauer. 2022. The Future of Mixed Reality Is Adaptive. *XRDS: Crossroads, The ACM Magazine for Students* 29, 1 (Sept. 2022), 26–31. https://doi.org/10.1145/3558191
[39] 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. https://doi.org/10.1145/3332165.3347945
[40] 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. https://doi.org/10.1109/VR46266.2020.00021
[41] 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. https://doi.org/10.1145/142750.142794
[42] 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. https://doi.org/10.1145/3613904.3642915
[43] 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. https://doi.org/10.1016/0003-6870(93)90080-S
[44] Mark R Mine. 1995. *Virtual environment interaction techniques*. Technical Report. UNC Chapel Hill CS Dept.
[45] 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. https://doi.org/10.1016/j.gmod.2023.101183
[46] 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. https://doi.org/10.1007/s11042-010-0525-z
[47] 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. https://doi.org/10.1109/JPROC.2020.2969687
[48] 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. https://doi.org/10.1145/2470654.2481307
[49] 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. https://doi.org/10.1145/3027063.3053213
[50] 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
[51] 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. https://doi.org/10.1145/3411764.3445426

[52] 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. https://doi.org/10.1145/1978942.1979348

[53] 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. https://doi.org/10.1145/3313831.3376438

[54] 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. https://doi.org/10.1109/TVCG.2022.3203096

[55] 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. https://doi.org/10.1007/1-4020-4179-9_46

[56] 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. https://doi.org/10.1109/ISMAR-Adjunct51615.2020.00072

[57] 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

[58] 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*. https://doi.org/10.1109/3DUI.2007.340783

[59] 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. https://doi.org/10.1145/3290605.3300310

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

[61] 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. https://doi.org/10.1109/TVCG.2023.3239951

[62] 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. https://doi.org/10.1145/3340764.3349525

[63] 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. https://doi.org/10.1145/3526113.3545632

[64] 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. https://doi.org/10.1145/3355089.3356544

[65] 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. https://doi.org/10.1109/TVCG.2020.3023606