IntenSelect+: Enhancing Score-Based Selection in Virtual Reality

Marcel Krüger, Tim Gerrits, Timon Römer, Torsten Kuhlen, and Tim Weissker

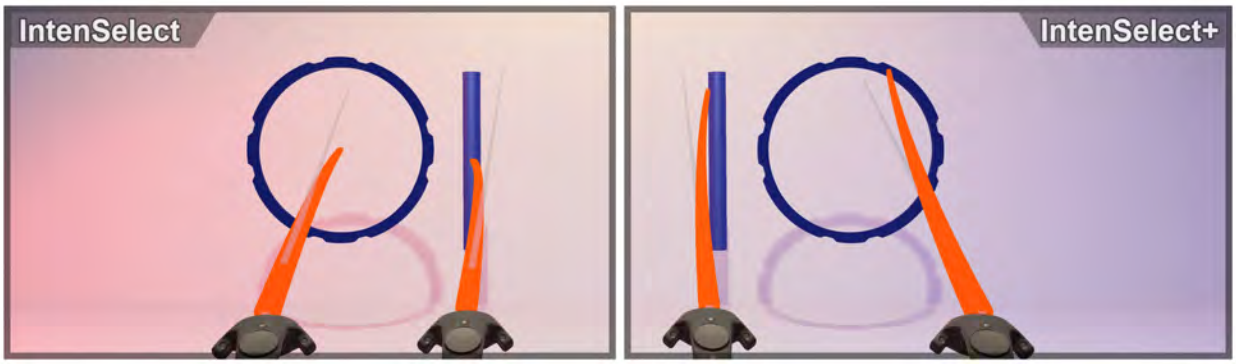

Fig. 1: We present *IntenSelect+*, an enhanced version of the temporal volume-based selection technique IntenSelect. Unlike its predecessor (left), IntenSelect+ is not restricted to representing objects as single points and thus always snaps to the closest point of an object to the selection ray (right). It also comes with an overhauled scoring function, which simplifies per-object parameterization.

Abstract— Object selection in virtual environments is one of the most common and recurring interaction tasks. Therefore, the used technique can critically influence a system's overall efficiency and usability. *IntenSelect* is a scoring-based selection-by-volume technique that was shown to offer improved selection performance over conventional raycasting in virtual reality. This initial method, however, is most pronounced for small spherical objects that converge to a point-like appearance only, is challenging to parameterize, and has inherent limitations in terms of flexibility. We present an enhanced version of IntenSelect called *IntenSelect+* designed to overcome multiple shortcomings of the original IntenSelect approach. In an empirical within-subjects user study with 42 participants, we compared IntenSelect+ to IntenSelect and conventional raycasting on various complex object configurations motivated by prior work. In addition to replicating the previously shown benefits of IntenSelect over raycasting, our results demonstrate significant advantages of IntenSelect+ over IntenSelect regarding selection performance, task load, and user experience. We, therefore, conclude that IntenSelect+ is a promising enhancement of the original approach that enables faster, more precise, and more comfortable object selection in immersive virtual environments.

Index Terms—Virtual Reality, 3D User Interfaces, 3D Interaction, Selection, Score-Based Selection, Temporal Selection, IntenSelect.

1 Introduction

Object selection is a fundamental task of 3D user interaction in virtual environments that is often the basis for further actions [8, 36]. While many selection methods exist, such as grasping, surface, indirect, and bimanual methods [8], pointing metaphors are still one of the most prominent selection categories. One popular pointing metaphor is the naïve raycasting method, which can be found in many VR applications. The precision of raycasting, however, decreases with distance based on the lever effect, which is particularly problematic for selecting smaller objects in dense environments. While prior work has presented several enhanced techniques to alleviate this issue [1, 22, 26, 31, 41], an auspicious approach considering the temporal context is *IntenSelect*, originally introduced by de Haan et al. in 2005 [12]. In a nutshell, IntenSelect updates a score for each object based on a scoring function and then selects the object with the highest score. De Haan et al. showed that this improves selection stability and usability over raycasting and volume selection if parameterized appropriately. However, the initial design of this technique was optimized for small spherical objects that converge to a point-like appearance, thereby limiting the generalizability of IntenSelect to more complex scenarios. Additionally, usability was reduced by complex dependencies between parameters and limited applicability when using per-object parameters.

In this paper, we present an enhanced version of IntenSelect called *IntenSelect+*. It enables the intuitive selection of objects, including complex shapes, using an improved scoring function and more flexible parameterization options on a per-object level. To evaluate our approach, we conducted a user study with 42 participants that compared IntenSelect+ to IntenSelect and naïve raycasting as baselines. To facilitate replication and accessibility to this technique, we published the executable of our user study and the implementation of IntenSelect+ as plugins for the Unreal game engine.

The design of our user study was particularly motivated by the desire to gain a more detailed understanding of the benefits and drawbacks of IntenSelect beyond the currently available insights in the literature. While the use of modern consumer-oriented VR hardware in our work is one factor contributing to this goal, we additionally focus on the explicit analysis of different types of object configurations that were previously shown to be potentially problematic for precise object selection [36]. As a result, this paper addresses the central research question of which selection technique is most beneficial for performing precise, fast, and comfortable selections in different object configurations. The contributions of our work to approach this question can, therefore, be summarized as follows:

- The introduction of *IntenSelect+*, an extended and more general version of IntenSelect that enables the selection of more complex shapes and offers an improved scoring function as well as

- *Marcel Krüger, Tim Gerrits, Timon Römer, Torsten Kuhlen and Tim Weissker are with the Visual Computing Institute at RWTH Aachen University. E-mail: {krueger, gerrits, t.roemer, kuhlen}@vr.rwth-aachen.de, me@tim-weissker.de*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

additional parameterizations

- Scientific evidence from a user study with 42 participants (i) validating previously shown benefits of IntenSelect over raycasting with modern HMD hardware, (ii) confirming that these benefits can be further strengthened with IntenSelect+, and (iii) providing guidance on interaction effects between the three techniques and different object configurations
- An exemplary implementation of IntenSelect+ as an open-source plugin for Unreal Engine to facilitate further research, development, and usage. [5]

Overall, our results encourage the use of IntenSelect+ over IntenSelect as well as raycasting and motivate future comparisons to alternative approaches for precise object selection in immersive virtual environments.

2 Background and Related Work

The selection of objects in 3D environments is a fundamental operation in virtual and extended environments. Numerous works have investigated accurate and efficient methods, focusing on various challenges in 3D scenes [24]. Nonetheless, finding new and improved selection techniques is still an active area of research, including specific challenges such as selecting text [44] or particle data [46], making use of additional technology such as eye tracking [39], or harnessing the support of deep neural networks [25]. While several taxonomies for the classification of selection techniques exist [7, 33], Argelaguet and Andujar [3] propose to classify techniques based on the selection tool and user behavior. As an in-depth discussion of the entire resulting design space and corresponding prior techniques is beyond the scope of this work, we refer the reader to further literature [6] and focus the following overview on ray-based pointing selection for single objects without changes to the scene, purposefully omitting touch/direct, grouped, or crossing selection approaches [20] for brevity.

Using a ray from a user's perspective, e.g., from a hand-held controller or the user's head or finger, that intersects objects targeted for selection has established itself as a quick and easy technique early in the development of virtual reality solutions [28]. However, shortcomings such as the "Heisenberg effect" [9, 42], i.e., the unintended movement of an input device when pressing a button, inaccuracy for small, moving, or distant objects in cluttered scenes [32], or a mismatch between ray- and gaze position [2] became imminent and required adaptations to the classic technique. Some examples summarized by Argelaguet and Andujar [3] propose adapting the classic ray with additional features such as a ray cast from the eye [4], applying two-handed pointing [29] or using two rays [43]. Furthermore, objects can be scaled based on their distance to the selection ray [1, 41], or additional views might be added [17, 21], allowing for a larger rendering of the objects. Another option is to change the selection ray to a volume, such as a cone [15, 26], which improves the selection of small and distant objects but might require the users to disambiguate objects within the selection volume [16]. To support this disambiguation, progressive refinement might be used [22, 38], or heuristics may determine the final target [37]. In 2020, Lu et al. presented the Bubble Ray [27] technique, which uses either the angular or Euclidian distance from the ray to the nearest object to make a selection. This approach, however, can still lead to unstable selections similar to regular raycasting as the Bubble Ray recomputes the distances to objects in every frame without incorporating temporal information. This can easily occur in dense scenes where objects have similar distances or in scenes where the projection of moving objects overlaps. Additionally, the maximum distance to the selected object was not limited such that selections can be involuntary, and the provided visual feedback can cover significant portions of the view. In contrast to these prior approaches, only a few works incorporated temporal information [30, 31, 36], i.e., the user's behavior, into the selection method, which can increase the stability of the selection. The most notable representative in this regard is IntenSelect by de Haan et al. [12], which the approach presented in this paper is built upon. Therefore, the following section summarizes the basic principles of this technique in more detail.

2.1 IntenSelect

IntenSelect by de Haan et al. [12], published in 2005, introduces spatial-temporal object scoring for volume selections. The idea is to let objects accumulate a score based on a weighted contribution factor calculated per frame. In each frame, the object with the highest score will be picked as the selected object. Therefore, the algorithm works in the following four stages:

1. Determine objects that are inside of a selection volume.
2. Calculate the score contribution that each object receives in the current spatial configuration.
3. Update each object's previous frame score with the current frame's score contribution.
4. Pick the highest-ranking object and give visual feedback to the user.

To calculate the score contribution for an object, first, the object's midpoint is projected into the local coordinate system of the selection cone. This allows calculating a distance-compensated angle between the center ray of the cone and the object by using a constant compensation factor k to prevent exponential growth, which results in

$$\alpha_{comp} = \tan^{-1}\left(\frac{d_{perp}}{(d_{proj})^k}\right)$$

with d_{perp} being the closest distance from the middle point of the object to the ray, and d_{proj} being the distance from the cone tip to the object in z-direction. This results in an overall scoring contribution function of

$$s_{contrib} = 1 - \frac{a_{comp}}{\beta}$$

with β being the opening angle of the cone. To retain temporal scoring information, the value is used to influence the overall score of the object, which gets updated each frame with the following formula:

$$s(t) = s(t-1) \cdot c_s + s_{contrib}(t) \cdot c_g$$

with c_s and c_g being the stickiness and snappiness constants respectively. These two constants influence how much of the score is changed between frames. Stickiness influences how much of the last frame's score is kept. Snappiness, on the other hand, how the current contribution is scaled before adding it to the total score.

The authors empirically tested the approach in a within-subjects design with eight participants and a 4×2 factorial design to gain first insights. In particular, the authors compared raycasting, simple volume selection, and IntenSelect in two different scenes. The first scene was a static scene in which small spheres had to be selected; the other scene had 16 spheres that moved through the scene periodically. The authors used a fixed pre-defined selection order and technique order of IntenSelect, volume selection, and finally, raycasting. To evaluate the obtained data, the authors chose to rely on descriptive analysis without performing inferential statistical tests to draw conclusions beyond the tested sample. The results indicated that volume selection was the fastest in the static setting, closely followed by IntenSelect and then raycasting. In the dynamic scene, however, IntenSelect resulted in the best selection time, followed by volume select and raycasting.

2.2 Discussion

IntenSelect behaves quite similarly to a standard raycasting approach. The user directly points in the direction of the object to be selected without any further necessary input but still gets significant support for selection. We, therefore, think it has the potential to be easily learnable while still providing significant benefits to the user. Additionally, it does not need any particular modifications of the scene or introduce visual change in the scene, i.e., through scaling, while using the interaction technique. This is especially important when the selection technique should also be applicable to tasks that do not allow changing the properties of the scene, i.e., for immersive visualization tasks.

However, in both the original [12] and a follow-up publication [11], the authors of IntenSelect mentioned several limitations of their approach and evaluation procedure, which particularly motivated us to carry out further investigations that we present in this paper.

First, the informal user study conducted by the authors only led to preliminary insights, and a formal and more rigorous user study may provide different outcomes and more detailed findings. Second, the system was only tested with a room-mounted display, namely a projection-based VR tabletop with an electromagnetically tracked stylus. As tracking technology steadily improved over the years, technological advances might lead to different results with regard to selection performance. Third, the evaluated scoring function is limited to point-like objects that are all weighted with the identical scoring function without options for individual customization.

Several recent comparative studies of IntenSelect [27,40], however, showed further promising results, especially in dynamic scenes, but they also reiterated inherent limitations of the approach, e.g., in highly cluttered scenes [41]. Due to a number of factors, we, therefore, revisited the original IntenSelect technique to discuss shortcomings and propose improvements to overcome them. Additionally, to technical improvements, we present the results of a 42-participant formal user study that provides insights into the original IntenSelect technique and our improved version.

3 IntenSelect+

After an initial analysis of the IntenSelect technique and early experimentation, we decided on three primary aspects that offer opportunities for improvement. Based on these aspects, we present an enhanced version called IntenSelect+ that is based on three core objectives:

1. Simplify the scoring function such that the adjustment of parameters becomes easier with predictable limits.
2. Allow objects to have comparable but independent per-object parameters for flexible scene control.
3. Extend the technique to allow for the selection of complex shapes.

We first discuss changes to the scoring function to address points 1 and 2 in Section 3.1. Afterward, we describe how the extended algorithm can be applied to more complex objects in Section 3.2.

3.1 Scoring Function

3.1.1 Limitations of IntenSelect's Scoring

One of the complexities when working with IntenSelect is the fine-tuning of the scoring behavior. Even though the score function $s(t)$ only has two direct parameters, namely c_g and c_s, hereafter defined as the tuple (c_g, c_s), the influence of and dependence between the two parameters is complex. This is primarily due to the recursive definition of the scoring algorithm, which results in a higher-order function where both parameters influence the overall score. This behavior is observable in Figure 2 (left); The object score can be observed for three different parameterizations when the object is fully hovered ($s_{contrib}(t) = 1$ for frame 0 to 30 and completely out of focus for frame 30 to 60. Let us consider the green curve as the baseline with $(0.9, 0.8)$. The red curve shows an adjustment of the stickiness parameter to $c_s = 0.85$ while keeping c_g constant. Even though the intention for such an adjustment would be to affect the score's decaying properties, one can observe that the shape of the growth phase is also affected by this change. Analogously, it can be seen in the blue curve that keeping c_s constant and setting $c_g = 0.5$ also influences the shape of the object's decay phase. This interdependence makes fine-tuning the stickiness and snappiness parameters of objects challenging.

Furthermore, as also visible in Figure 2, the complex interdependence of c_s and c_g also leads to varying maximum values of the scoring function, which is especially complex and limiting in situations where per-object parametrization is needed. While the function is bound, the limit and temporal behavior depend on the concrete values of c_s and c_g. Imagine a scenario in which two objects O_1 and O_2 have different parameters, e.g., $c_{O_1} = (0.9, 0.95)$ and $c_{O_2} = (0.5, 0.95)$, which means that only the growth factor should be different between both objects. Due to the described interdependence, however, both values approach a different limit $\hat{s}_i$, with $\hat{s}_{O_2} < \hat{s}_{O_1}$. This can lead to situations as depicted in Figure 2 (right), where even though O_2 is directly on the center ray while $O1$ is at some distance, the wrong object is selected. Due to the different limits of the score, object O_1 will always be selected even though its current contribution $s_{contrib,O_1}(t) = 0.6$ is smaller than the one of O_2 with $s_{contrib,O_2}(t) = 1.0$, thus O_2 is virtually unselectable. Since this interaction effect is only amplified if more objects with distinct parameters are introduced, the complexity of balancing the parameterization quickly becomes infeasible.

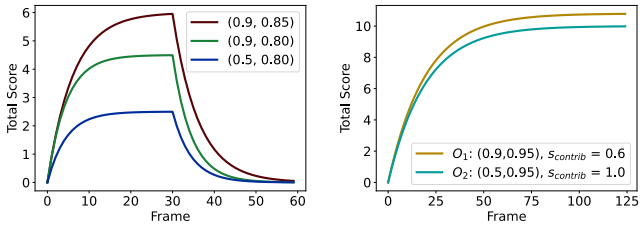

Fig. 2: IntenSelect's original scoring function with varying stickiness and snappiness parameters. Left: Changing one parameter directly influences the overall behavior. Right: Different per-object parameters can lead to one object having a lower score albeit having more contribution.

3.1.2 Enhanced Scoring Function of IntenSelect+

Based on the described limitations, we revised the scoring function with two goals: (i) simplify the parameter-dependent behavior of objects, (ii) establish a common upper limit to make scores of objects with different parameters comparable.

To achieve goal (i), we first split the scoring function into two cases: One case that controls the score's growth and one case that controls decay. For this, we use the parameters c_g for the growth-rate and c_d for the decay-rate. The parameters only influence the behavior of their respective cases and changing one does not influence the behavior of the other case. To achieve goal (ii), we decided to let the overall score always approach $s_{contrib}(t)$. As $s_{contrib}(t)$'s limits are not object-specific but global, the overall score $s(t)$ is also globally bound and thus comparable across objects. Additionally, we sought a formulation that is stable against small variations in the score contribution while emphasizing large and enduring change. We achieve this by using an interpolated difference between the current score $s(t)$ and $s_{contrib}(t)$ that weighs how much the score should be changed:

$$s_{\delta interp} = (x \cdot s(t-1) + (1-x) \cdot s_{contrib}(t)) - s(t-1) \qquad (1)$$

with $x \in [0, 1)$ being the interpolation factor. The interpolation factor x can be used to be more sensitive to the current contribution or the existing score. We permanently set $x = .5$ in our current implementation to weigh past and current scores the same. In each frame, the total score $s(t)$ of an object can then be calculated as:

$$s(t) = \begin{cases} s(t-1) + s_{\delta interp} \cdot \Delta_t \cdot c_g, & \text{if } s(t-1) \leq s_{contrib}(t). \\ s(t-1) + s_{\delta interp} \cdot \Delta_t \cdot c_d, & \text{if } s(t-1) > s_{contrib}(t). \end{cases} \qquad (2)$$

with Δ_t being the frame deltatime that enables framerate-independent scoring, which was not possible with the original scoring function. The first case is used when the score should rise; therefore, it is only dependent on c_g. Respectively, the second case is only dependent on c_d and is used if the score should be lowered. Assuming $s_{contrib}(t)$ stays constant as well as $\frac{1}{1-x} \cdot \Delta_t \cdot c_d < 1$ and $\frac{1}{1-x} \cdot \Delta_t \cdot c_g < 1$, the difference gets smaller over time; thus, $s(t)$ will converge to $s_{contrib}(t)$ if $c_g, c_d \in (0, \frac{1-x}{\Delta_t})$. As $s_{contrib}(t) \in [0, 1]$, it therefore follows that $s(t) \in (0, 1)$. Hint: $s_{contrib}(t)$ equals 0 if the object is outside the selection cone and increases to 1 when the middle ray is on the object (c.f. 3.2.1).

While the new formulation still has the same number of parameters as the orignal technique, tuning the behavior is simplified as they are independent from each other. This is in contrast to the original implementation, where changing one parameter would influence both stickiness and snappiness. Additionally, the new formula leads to a consistent positive correlation for both c_d and c_g, meaning that a higher value means that the score decays or grows faster. In the original formula, c_s meant how much of the old score is retained, which led to a negative correlation. Overall, we believe that this leads to a simplified process of finding the correct parameters, therefore addressing goal (ii).

This results in a flexible scoring function that is easier to parameterize and makes score comparisons between objects sensible even if distinctive per-object parameters of c_g and c_d are used.

3.2 Selection Primitives

3.2.1 Limitations of IntenSelect's Selectable Objects

The second major motivation for IntenSelect+ was enabling the selection of more complex shapes. This is especially important when the use case is not restricted to small sphere-like objects and, therefore, contains objects with larger spatial extents. In particular, objects that have small cross-sections depending on the view or in certain dimensions can be hard to select without additional help. Prominent examples in this regard include lines since they are not adequately represented by a single point.

The authors of the original IntenSelect method suggest three different methods to deal with larger objects [11], namely, using the midpoint, using a custom-defined point, and using the last point where a direct ray intersection occurred. However, the first two do not fix the limitation mentioned above, since finding a single point that is representative of a whole object and is reachable at all times is hard to solve in various cases. The third approach has the inherent restriction that only the last point of intersection will be considered as the interaction point. Thus, the user cannot make use of IntenSelect before the first intersection occurs. Additionally, the computed distance to the object is only correct if the vector representing the shortest distance from the last intersection point to the central axis of the selection cone is perpendicular. At worst, the distance is significantly overestimated in cases where the ray is close to the object but far away from the last intersection point. Lastly, situations where the user approaches the object from a different side, e.g., the other side of the object, can lead to abrupt changes in the score and selected objects.

3.2.2 Additionally Selectable Objects with IntenSelect+

To go beyond the restriction of objects to a single point, we suggest representing objects by one or multiple geometric primitives for which the computation of their distance to the selection ray can be performed at low cost. These geometric primitives are introduced as an exact calculation of the distance between a ray and arbitary shapes is infeasable. To realize this in our implementation, we move all scoring-related functionality into the logic of the object, which means that each object is responsible for calculating its own score. This approach results in two significant advantages: First, it allows us to define c_d and c_g as part of an object's property instead of it being a global property of the IntenSelect algorithm. Thus, c_d and c_g can be additionally dependent on object properties such as speed, volume, etc. Second, it enables the use of shape-dependent per-object functions that are used for the computation of $s_{contrib}(t)$, which leads to more versatility than the reduction to the object's center point in the original IntenSelect algorithm.

To identify objects that fall into the selection cone in a given frame, we use a performant sphere-cast with its main axis along the middle ray and radius set to the radius at the endpoint of the cone to find candidate objects. The sphere-cast used in our implementation employs highly optimized acceleration structures to efficiently query the scene at low costs. This is only an optimization step to reduce the number of distances that have to be calculated in the scene and can be omitted. Candidate objects are then given the cone origin, direction, and cone angle, which are then used as input parameters to the object's scoring function. The algorithm finds the closest point on the geometric primitive representing this object and then calculates the score contribution

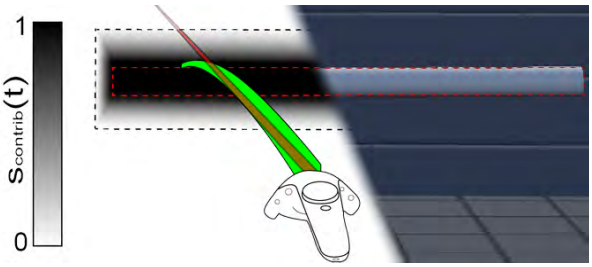

Fig. 3: Signed distance function of a cylinder. The function evaluates to 1 inside of the cylinder and 0 on the edge of the primitive.

$s_{contrib}(t)$ based on the shortest Euclidian distance between the cone's middle ray and the primitive. Primitives for which this calculation was implemented will be presented in the following paragraphs. As the exact formulas are basic ray-to-primitive calculations we kindly refer the reader to literature [13] and our code. $s_{contrib}(t)$, therefore, essentially behaves like a signed distance function with a value between 0 and 1 (c.f. Figure 3). The function evaluates to 1 inside of the object, as well as on its surface, and then decreases to zero 0 at the edge of the selection cone. The exact distance at which the SDF evaluates to zero is defined by the cone angle and compensation factor k, which shapes the cone analogously to the original technique. If an object is represented by more than one primitive, the highest individual score will be returned as a representative score for the object.

In our current implementation, we provide exact scoring functions for points, lines, surfaces, and volumes, which are motivated on a conceptual level in the following.

Points We implemented two kinds of point functions: simple points and multipoints. Simple point handles are analogous to the original implementation in that they represent a single 3D position on an object. Multipoints, on the other hand, are groups of points that define locations on objects, where each point gets an individual score, but only the highest subscore is reported as the object's score. This allows to define multiple handles per object without overrepresenting the object in the selection algorithm.

Lines/Surfaces Lines represent 1D primitives that are useful to allow selection on objects that are still thin enough that they are still an adequate approximation, e.g., thin cylindrical objects. For surfaces, we currently provide implementations for disks and rectangles. Even though these primitives represent surfaces, they can still be used to approximate 3D objects. While using this approximation decreases the accuracy of the object representation for selection, they generally reduce the complexity of the score calculation as calculations become simpler.

Volumes For large objects, however, the approximation becomes noticeable and limiting; therefore, the object can also be represented via a volume primitive. In our implementation, we currently provide scoring functions for spheres, cubes, and cylinders. For 3D objects, we allow backface culling where applicable to reduce computational demand.

Compound Shapes Our implementation allows the grouping of different primitives into a compound. This enables the approximation of various shapes due to the combination of approximate objects. Analogously to the multipoints, only the largest subscore will be used as the object score, effectively behaving like a union operation.

For 2D and 3D objects, the user can additionally decide to enable IntenSelect+ for the outline of the primitive, thus allowing us to also realize shapes like circles or wireframe boxes to create more complex representations.

The definition of IntenSelect+ allows the user to easily replicate IntenSelect behavior by only using single points and raycasting by setting both c_g and c_d sufficiently high and reducing the cone to a ray via the opening angle.

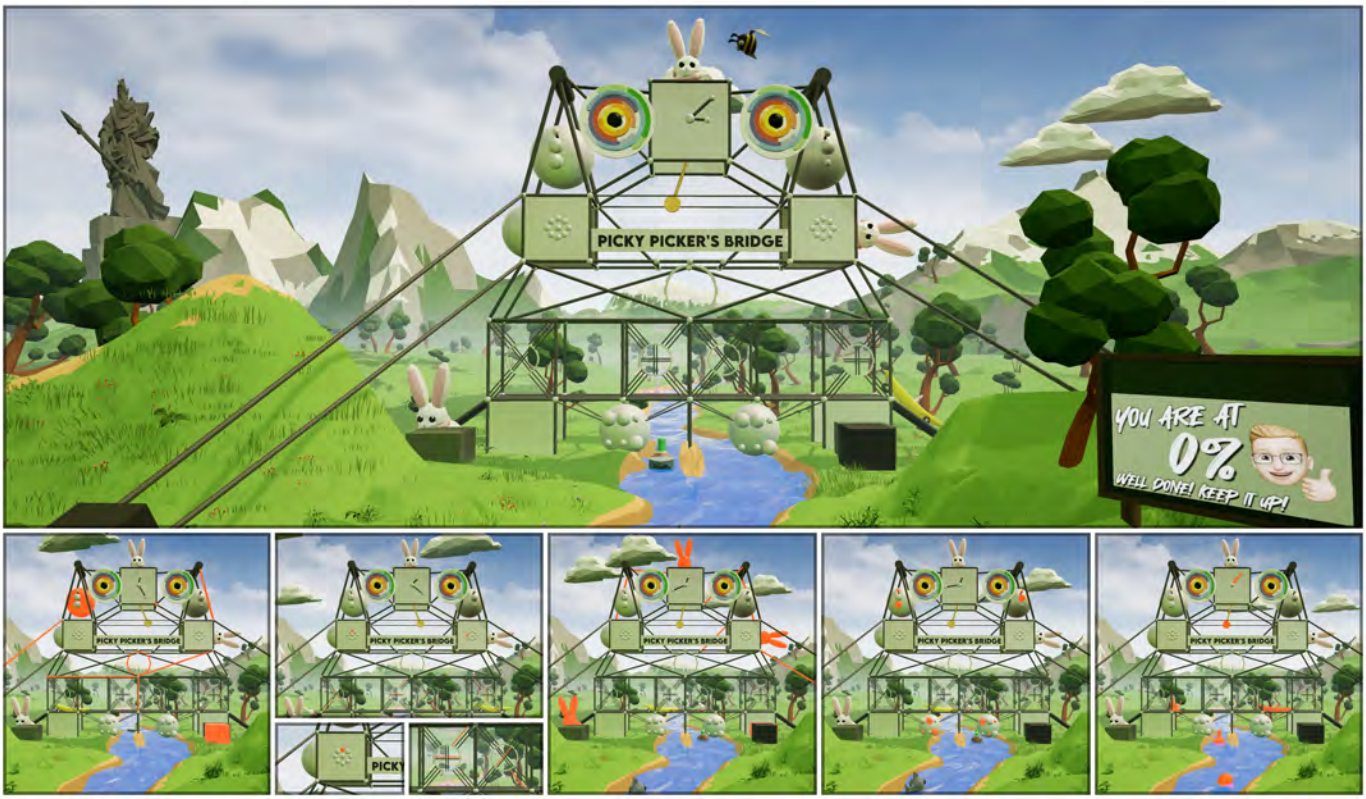

Fig. 4: Scene in which users were asked to select objects in our user study. In each tile, all elements of the respective object configuration category are highlighted in orange. In the actual study, only one element that had to be selected was highlighted at a time. Overall, all of the highlighted elements had to be selected, resulting in 50 selections per technique. *This depiction was slightly altered for anonymization.*

4 Empirical Evaluation of IntenSelect+

We conducted a formal user study with two independent variables to quantify the benefits and drawbacks of using IntenSelect+ on different object configurations. The first independent variable was *Selection Technique*, for which we decided to compare IntenSelect+ (IS+) to the baselines given by the unaltered version of IntenSelect (IS) as well as conventional raycasting (RC). The second independent variable was *Object Configuration*, for which we decided to compare technique usage with five types identified by prior work as well as our experience. These included the three challenging configurations highlighted by Steed [36] of objects that are close to other objects (CT), partially occluded objects (OC), and small objects in front of other large objects (SL), which we complemented with isolated stationary objects as a baseline (ST) as well as moving objects (MV). The details of our study are presented in the following.

4.1 Apparatus

The study was performed in our virtual reality lab, where participants performed the experiment in a stationary standing position with a head-mounted display. For this, an HTC Vive Pro 2 with respective HTC Vive controllers was used. The HMD ran with a resolution of 4896 by 2448 pixels and a constant refresh rate of 120 Hz such that reprojection never occurred. The positions and rotations of the HMD and controllers were tracked throughout the experiment by one frontal-facing Lighthouse 2.0 base station. The application was run on a Windows desktop PC with an Intel Core i9-10900X processor, an NVIDIA Geforce RTX 3090 graphics card, and 32GB of DDR4 RAM.

4.2 Participants

The study was done by 42 participants, of which 21 identified as male, 18 as female, and three as diverse. The age of the participants ranged from 19 to 46 ($M = 25.95$, $SD = 5.486$). Participation in the study required normal or corrected-to-normal vision and fluent English skills. Recruitment was done locally at the university campus through various communication channels. In addition to age and gender, participants self-reported their proficiency with virtual reality and gaming. For VR experience, 2 participants stated they have no experience, while 14, 15, and 11 participants rated themselves as beginner (used a few times before), advanced (used several times before), and expert (regular usage), respectively. For game experience, 1 participant stated to have no experience, while 12, 6, and 23 participants rated themselves as beginner, advanced, and expert users, respectively. All participants successfully completed the full experiment, leading to data points for $N = 42$ participants for statistical analysis.

4.3 Task

Our experimental task required participants to select a range of different objects in a virtual environment consisting of a low-poly landscape with mountains, hills, trees, foliage, and a river.

To provide suitable targets for the selection task, a bridge-like structure with different geometric primitives was placed into the virtual landscape. The user's task was to select elements in the bridge that were progressively highlighted by a distinctive red color. Users were instructed to select only the indicated element and prioritize correctness over speed while still trying to select them as fast as possible with the respective technique. The selection was, however, not restricted to the indicated elements such that false selections could occur. Additionally, visual and acoustical signaling was used to indicate the element that had to be selected next only when the new element became active. Once a selection was made, the indicated element was reset to the base color.

The selection was performed by indicating the desired element with the currently used technique and then pressing the trigger button, which started a timer of 500ms, during which the selection should be maintained. This dwell time was introduced to reduce the number of erroneous selections by giving the user the chance to abort by releasing the trigger in case the wrong element was selected. The elapsed time

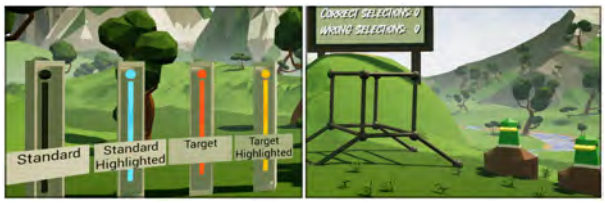

Fig. 5: Tutorial scene used in our study, featuring an overview of element colors (left) and example objects for practicing (right).

was constantly visible to the user via a circular progress bar next to the selected point.

Per selection, the user had a maximum of 15s to complete the selection. This was introduced to prevent overwhelming the user in case of complex selections and additionally to force them to make selections and not take too much time. In case of timeouts, i.e., no successful selection within the time limit, we chose to count the selection as a false selection. After each selection, a 2s inter-repetition delay was used before the new element was indicated to let the user refocus on the task and give the opportunity to mentally conclude the previous selection. The end of each repetition, either through successful or wrong selection or timeout, was indicated by the same acoustical signal to not indicate the correctness of the selection. The overall progress for each selection technique was communicated via a billboard embedded in the environment.

For each of the above-introduced object configurations, we selected ten representative objects in the scene such that each of them fell into one and only one of the presented categories. As a result, users had to select a total of 50 different objects in the scene. While these objects were the same for all three tested selection techniques, their order of appearance differed to minimize order effects during the repetitions. However, we ensured a balanced presentation of the five object configurations using a balanced Latin Square design to prevent an overrepresentation of a particular object configuration in a particular phase of the study. For each technique, a unique permutation of the Latin Square was used. Figure 4 shows the scene with each object configuration highlighted in the scene respectively. For our study, we empirically determined parameters to produce similar temporal behavior. For IntenSelect, we chose $c_s = .87$ and $c_g = .13$. For IntenSelect+, we chose $c_d = 10$, $c_g = 15$. Both used $k = 0.8$ for all objects.

4.4 Procedure

Users were welcomed and signed an informed consent outlining the collection and usage of the data that was recorded in the study. Users then drew a random ID that was used for pseudonymization of the results and asked to fill out a questionnaire regarding their demographic information. Users were then shown the VR hardware to be used, and the equipment was adjusted to fit comfortably. Users were directly shown a tutorial environment to let them familiarize themselves with the hardware. When ready, participants pressed a button that started a video playback in the virtual environment describing the study setup and the task to be performed. Before continuing with the study task, participants were given the opportunity to ask questions to the experimenter.

After this general introduction phase, the following procedure was repeated for all three techniques. First, the user was shown an introduction video describing the technique to be used and had the opportunity to try it out in the reduced tutorial environment without limitations on the number of trials or time (c.f. Figure 5). Users were able to continue to the recorded task when they felt ready after at least five successful selections via a button press on their controller.

The scene then changed to the bridge environment, where the user could again indicate readiness via a button press to start the task sequence. After completing the required 50 selections, the user was asked to take the HMD off and fill in a questionnaire, which consisted of a single-value discomfort score [34], the Raw TLX as a simplified version of the NASA-TLX with equal subscale weighting [18, 19], and the user experience questionnaire (UEQ) [23].

After all three techniques were presented and the respective questionnaires were filled out, a post-study questionnaire was provided to the participants where they were asked to rank their favorite, second favorite, and least favorite techniques and provide any comments on the study. The whole procedure took between 45 and 60 minutes per participant.

4.5 Hypotheses

Given our intentions of developing an improved version of IntenSelect, we hypothesized that **(H_1) the selection performance (i.e., mean selection time and error) is lowest for IS+, followed by IS in second and RC in third place**. We also hypothesized that these overall performance benefits manifest themselves within each type of object configuration by stating that **(H_2) IS+ performs best within all tested object configuations**. Regarding interaction effects between selection technique and object configuration, we expected that **(H_3) RC would perform worst for the CT object configuration** as the lever effect of raycasting can easily lead to the selection of neighboring objects in close proximity. Furthermore, we expected that **(H_4) IS performs worst for the OC object configuration** as the midpoint of larger objects lies behind other objects in this case. Therefore, it might be harder and more confusing for the user to select these objects since they have to point directly toward the occluding object. We also hypothesized that **(H_5) IS requires more hand rotations than the other techniques** as the user has to rotate towards the midpoint of each object instead of selecting the point closest to the ray.

With respect to the questionnaire data, we expected that **(H_6) IS+ requires the lowest task load** based on its reduced sensitivity for jitter and less required anticipation compared to IS. We also hypothesized that these positive effects, as well as the overall improvements introduced by IS+, are reflected on the User Experience Questionnaire by stating that **(H_7) IS+ performs better than the other two techniques on all six subscales of user experience**.

We did not hypothesize differences between techniques and categories regarding user well-being as measured by the discomfort score, which is due to the identical amount of visual flow introduced by each condition. Based on our preference ranking, however, we expected that **(H_8) IS+ is generally preferred over the other two techniques**.

5 Results

Guided by our hypotheses, we analyzed the data of our user study with *IBM SPSS Statistics 28*. For hypothesized overall differences between techniques, we conducted one-way repeated-measures ANOVAs to test for statistically significant effects. For hypotheses involving both technique and object configuration, 3×5 factorial repeated-measures ANOVAs were conducted instead to identify main and interaction effects. To prevent an overreliance on p-values, we computed the effect sizes η_p^2 for each ANOVA and applied the threshold values of .01, .06, and .14 suggested by Cohen [10, pp. 285–287] to categorize small, medium, and large effects, respectively. Variables without a hypothesis were analyzed purely descriptively.

Two key requirements for the validity of repeated-measures ANOVAs are a normal sampling distribution and equal variances of the differences between conditions, also referred to as sphericity. While our sample size of $N = 42$ is sufficiently large to carefully assume a normal sampling distribution based on the central limit theorem [14, pp. 170–172], we tested the assumption of sphericity using Mauchly's tests and reported Greenhouse-Geisser corrected values and the correction factor ε when it was violated.

For more detailed insights than the overall analyses of ANOVAs provide, we conducted follow-up paired-sample t-tests to identify pairwise differences. In these cases, we applied Bonferroni corrections of the p-values to counteract the risk of inflated error rates. We also computed the effect size Cohen's d with the threshold values of 0.2, 0.5, and 0.8 for the above-mentioned effect categories [10, pp. 24–26]. The data files of our user study are provided as supplemental material to this publication for additional clarity.

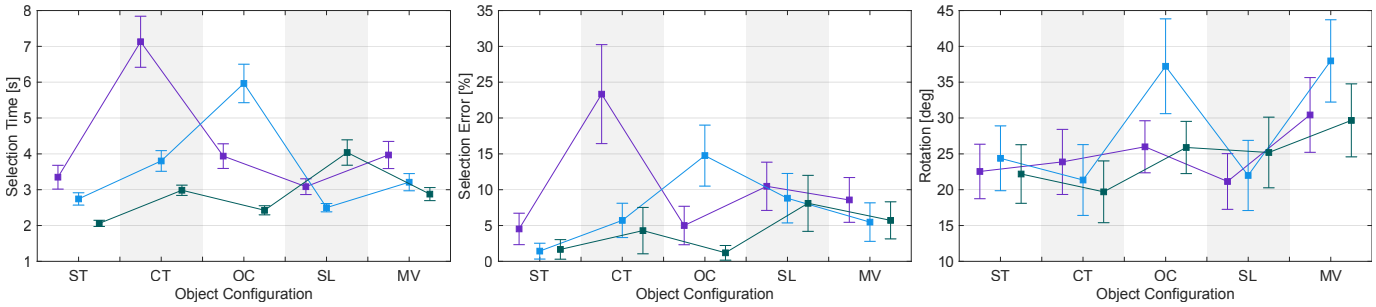

Fig. 6: Graphical depiction of interaction effects between the two independent variables *Selection Technique* and *Object Configuration* regarding mean selection time (left), mean selection error (center), and mean rotation angle (right). Each plot shows the arithmetic means of the three techniques split by object configuration, with error bars representing the 95% confidence intervals of each mean. The colors represent Raycasting (RC), IntenSelect (IS), and IntenSelect+ (IS+). The tested object configurations were isolated stationary objects (ST), objects that are close to other objects (CT), partially occluded objects (OC), small objects in front of large objects (SL), and moving objects (MV).

5.1 Inferential Tests on Logging Data

Boxplots of the measured selection times, error rates, and accumulated controller rotations are given in Figure 8 and supplemented by inferential analyses in the following:

Selection Time The assumption of sphericity was violated for the main effect of technique ($\varepsilon = .777$), the main effect of the object configuration ($\varepsilon = .728$), and the interaction effect ($\varepsilon = .498$). There was a significant main effect of technique ($F(1.555, 63.741) = 66.434$, $p < .001$, $\eta_p^2 = .618$), a significant main effect of the object configuration ($F(2.913, 119.439) = 96.765$, $p < .001$, $\eta_p^2 = .702$), and a significant interaction effect ($F(3.983, 163.300) = 100.026$, $p < .001$, $\eta_p^2 = .709$), all with large effect sizes. Post-hoc tests revealed significant differences between all pairs of techniques (all $p < .001$), with a medium effect between RC and IS ($d = 0.713$) and large effects for the other comparisons (both $d > 1.4$). The interactions between technique and object configuration are depicted in Figure 6 (left).

Selection Errors The assumption of sphericity was violated for the main effect of the object configuration ($\varepsilon = .811$) and the interaction effect ($\varepsilon = .590$). There was a significant main effect of technique ($F(2, 82) = 18.112$, $p < 0.001$, $\eta_p^2 = 0.306$), a significant main effect of the object configuration ($F(3.246, 133.082) = 14.804$, $p < 0.001$, $\eta_p^2 = 0.265$), and a significant interaction effect ($F(4.722, 193.616) = 16.613$, $p < 0.001$, $\eta_p^2 = 0.288$), all with large effect sizes. Post-hoc tests revealed significant differences between all pairs of techniques (all $p < .02$), with a small effect between RC and IS ($d = 0.471$), a medium effect between IS and IS+ ($d = 0.523$), and a large effect between RC and IS+ ($d = 0.836$). The interactions between technique and object configuration are depicted in Figure 6 (center).

Accumulated Controller Rotations The assumption of sphericity was violated for the main effect of object configuration ($\varepsilon = .679$) and the interaction effect ($\varepsilon = .561$). There was a significant main effect of technique ($F(2, 82) = 5.671$, $p = .005$, $\eta_p^2 = .122$), a significant main effect of the object configuration ($F(2.714, 111.293) = 68.735$, $p < .001$, $\eta_p^2 = .626$), and a significant interaction effect ($F(4.491, 184.127) = 13.009$, $p < .001$, $\eta_p^2 = .241$), all with large effect sizes. Post-hoc tests revealed significant differences between RC and IS ($p = .045$, $d = 0.391$) as well as IS and IS+ ($p = .001$, $d = 0.584$), with small and medium effect sizes, respectively. The comparison between RC and IS+ was not significant ($p = 1.000$, $d = 0.029$). The interactions between technique and object configuration are depicted in Figure 6 (right).

5.2 Inferential Tests on Questionnaire Data

Task Load The task load scores were significantly affected by technique with a large effect size, $F(2, 82) = 45.882$, $p < .001$, $\eta_p^2 = .528$. Post-hoc tests revealed significant differences between all pairs of techniques (all $p < .001$), with a medium effect between IS and IS+ ($d = 0.692$) and large effects for the comparison of RC and IS ($d = 0.825$) and the comparison of RC and IS+ ($d = 1.426$).

	Overall Test			Post-Hoc Tests		
	F	p	η_p^2	d(RC, IS)	d(RC, IS+)	d(IS, IS+)
Attractiveness	63.086	<.001	**.606***	**0.813***	**1.957***	**0.835***
Perspicuity	14.087	<.001	**.256***	**0.571***	0.180	**0.722***
Efficiency	42.662	<.001	**.510***	**0.629***	**1.422***	**0.835***
Dependability	15.092	<.001	**.269***	0.028	**0.801***	**0.692***
Stimulation	46.055	<.001	**.529***	**1.018***	**1.394***	**0.449***
Novelty	75.745	<.001	**.649***	**1.509***	**1.538***	0.050

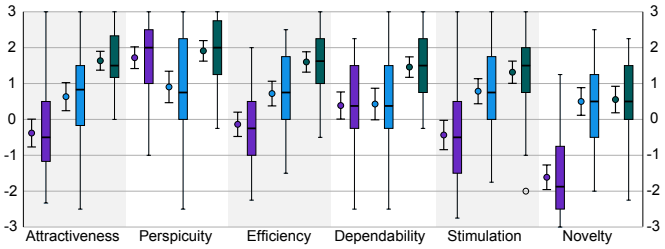

Fig. 7: Top: Results of the inferential tests applied on the six subscales of the User Experience Questionnaire (UEQ), in which statically significant effect sizes are highlighted in bold and marked with an asterisk. Bottom: Boxplots illustrating the distribution of UEQ subscale scores. Each individual boxplot is complemented by a circle indicating the arithmetic mean as well as the 95% confidence interval of the mean. The colors represent Raycasting (RC), IntenSelect (IS), and IntenSelect+ (IS+).

User Experience The results of the inferential tests applied on the six subscales of the User Experience Questionnaire are given in Figure 7. Overall, all subscales showed significant differences between techniques with large effect sizes. Significant differences were also found for most of the pairwise follow-up comparisons, with the exceptions of perspicuity for the comparison of RC and IS+ ($p = .753$), dependability for the comparison of RC and IS ($p = 1.000$), and novelty for the comparison of IS and IS+ ($p = 1.000$).

5.3 Further Descriptive Analyses

Discomfort The discomfort scores ranged from 0 to 9 for RC ($M = 2.26$, $SD = 2.359$), from 0 to 9 for IS ($M = 1.55$, $SD = 1.978$), and from 0 to 8 for IS+ ($M = 1.45$, $SD = 1.811$). The median score was 2 for RC and 1 for the other two techniques.

Technique Preference IS+ was the overall most favored technique, with 35 and 7 participants ranking it first and second, respectively. In contrast, RC was the overall least favored technique, with 10 and 32 participants ranking it second and third, respectively. The preference rankings of IS were situated between these extremes, with 7, 25, and 10 participants ranking it first, second, and third, respectively.

5.4 Interpretation and Discussion

Overall, our results paint the unanimous picture that IS+ offered several advantages over IS, which in turn was beneficial over RC, as expected based on evaluations in prior work.

5.4.1 Objective Measurements

It can be observed that the mean selection time with regard to the selection technique is the fastest with the IS+ technique. In more detail, we observed that IS+ < IS < RC with regard to the mean selection time. Thus, we can confirm $\mathbf{H_1}$ based on the statistical tests reported in Section 5.1, which show that the differences in the selection times are significant with effect sizes ranging from medium to large effects. While the overall improvement in selection time is a relevant result, it is especially interesting to look at specific object configurations to gain further insights into the strengths and weaknesses of each technique.

Taking the selection time and error rate into consideration for all object configurations, we can observe that IS+ performs best in all categories except for SL selections. The error rate of IS+ is always better or at least comparable to the other two techniques. While the selection time achieved for ST, CT, OC, and MV is the best with the IS+ technique, it is different for SL selections. In this case, IS+ performs the worst of all selection techniques, with RC in the middle and IS performing best. Thus, we reject $\mathbf{H_2}$ as we identified a weak point of IS+ in SL selections, requiring further improvements. The problem in the interaction between IS+ and SL is that the small object is completely engulfed in the large when projecting the objects onto the cone intersection. This means that the larger object will receive the maximum score contribution if the user is close to the small object but also if the user is directly on it. Therefore, the smaller object can only reach an equal score if and only if the user directly hits the small object. Consequently, the advantages of IS+ are completely counteracted for the small object, as seen in the results. The same problem occurs for IS only if both midpoints are lying exactly on the middle ray. There are multiple possible solutions for this issue. One option would be to prevent score increases of other objects if an object is directly hit, and another option is to make use of the per-object-scoring to give the smaller object more aggressive parameters to gain selection faster and stick around longer, i.e., increase c_g and decrease c_d. However, in the case of per-object scoring, it would be difficult to find a solution that is generally applicable as occlusion problems are always dependent on the user's exact viewpoint.

The biggest differences in selection performance can be observed in CT-type object configurations, with RC for CT selections resulting in the slowest selection times overall. We could observe in our study that users had difficulties selecting the correct object, as small deviations in the controller orientation resulted in the indication of surrounding objects that were closeby. Thus, user slowed down their selection to make sure that the correct object was selected before completing the selection. Figure 6 (center) shows that even though selection times were slower compared to other techniques, it still resulted in the lowest correctness score for any combination. We observe significant differences for both selection time and selection error with small to large effect sizes for all pairwise comparisons. This confirms our hypothesis $\mathbf{H_3}$ that the selection performance for RC in CT type selections is the worst among all selections.

For OC-type selections, it can be observed that the selection time is the worst for the IS technique, while the best time can be achieved with IS+ with the time of RC situated in between. Combined with the increased error rate for IS when doing OC selections, we can see that the performance of IS is the worst in this object configuration. Based on significant pairwise differences with small to large effects for all combinations, we confirm hypothesis $\mathbf{H_4}$. In the study, we observed that users found it very hard to estimate where the middle point of the object lies. In the OC case, the middle point was occluded to varying degrees, i.e., the midpoint was closer or further away from the visible sections. This led to the behavior that users generally tried to follow the shape of the object and search for the object's midpoint instead of directly moving to it. The disadvantage was observed further in the free comment section that users could use at the end of the study. 21 of 42 participants explicitly commented that this was a challenging object configuration and needed a lot of concentration.

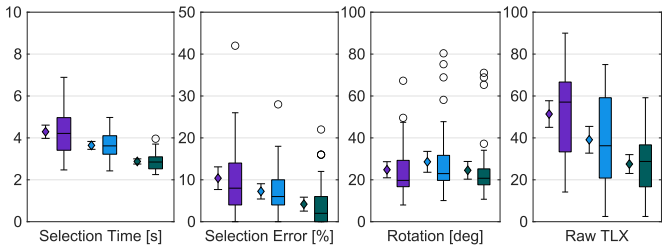

Fig. 8: Boxplots showing selection time, error rate, controller rotation, and Raw TLX values for the techniques Raycasting (RC), IntenSelect (IS), and IntenSelect+ (IS+). Next to each boxplot, a diamond represents the arithmetic mean with 95% confidence intervals.

Considering the amount of controller rotation needed per selection, it can be seen in Figure 6 (right) that IS requires more hand rotation compared to RC and IS+. This is due to the limitation of IS that the user has to always rotate to the midpoint of an object even if it has large dimensions in one or more axes. When considering a large line, for example, the user can select the object as soon as the selection line or volume reaches the border of the object, and the user can stop rotating. This is especially pronounced in the OC and MV object configurations, as the user will stop the rotation when selecting as soon as possible and does not have to reach or track the midpoint. The statistical analysis supports this assumption as there are significant differences between RC and IS as well as IS and IS+ with small and medium effect sizes, while there is no difference between RC and IS+. This indicates that in terms of the required amount of controller rotations, RC and IS+ behave similarly. We thus accept $\mathbf{H_5}$ as IS requires more hand rotations compared to both other techniques.

5.4.2 Subjective Measurements

When taking subjective measures into account, it can be observed that the Raw TLX score is the highest for RC, second highest for IS, and the lowest for IS+. Statistical analysis revealed a significant difference with an overall large effect size for the task load scores with a medium effect between IS and IS+ as well as large effects for IS and IS+ compared to RC. This confirms $\mathbf{H_6}$ that IS+ results in the lowest task load. In more detail, it can be seen that when considering all techniques, RC generates the highest task load while IS is situated between RC and IS+.

The statistical analysis of the UEQ revealed that all six subscales show significant differences with regard to the rated techniques with large effect sizes. For further discussion, we split the UEQ into three semantic units as described by Schrepp et al. [35]. Pragmatic qualities represent goal-oriented measurements that primarily evaluate effectiveness. Hedonic qualities, on the other hand, represent aspects that are not directly goal-oriented. Both categories, however, influence the overall attractiveness that the user assigns to a product, allowing one to rate one system as more attractive than another system if they are as effective but one is more enjoyable to use.

Pragmatic Qualities Pragmatic qualities in the context of the UEQ are perspicuity, efficiency, and dependability. For perspicuity, we can observe significant differences between RC and IS as well as IS and IS+, with medium effect sizes, while there is no significant difference between RC and IS. In the overall order in terms of perspicuity, IS+ has the best score, followed by RC and IS. As perspicuity describes how easy it is to learn or get familiar with the system, no significant difference between RC and IS+ could hint that users felt a similar level of complexity and learning for both techniques. This is additionally supported by the positive absolute values for RC and IS+, which give an indication that users are able to learn the system well. The significant differences to IS, on the other hand, indicate that users had a harder time getting familiar with this condition. We speculate that due to the

technique, the means of interaction with the object slightly changes, especially in cases like OC object configurations where pointing through other objects to select an object is rather unintuitive. Therefore, IS might require a larger amount of familiarization compared to other techniques.

Once learned, however, users seem to find IS more efficient than RC, while IS+ gets the best scoring of all three techniques, which is represented by the UEQ's efficiency subscale with a medium effect size when comparing RC to IS and large effect sizes between RC and IS+ as well as IS and IS+. This indicates that the support provided by IS and IS+, compared to the straightforward input via RC, helps the user in completing the task faster and more precisely. Furthermore, it shows that the user feels that their efficiency is further increased when comparing IS and IS+.

This is also reflected in the dependability subscale of the UEQ, which represents the degree of control the user feels when using the system. We can observe the same ordering of RC < IS < IS+. No statistically significant difference between RC and IS could be observed, while the pairwise comparison of RS and IS+, as well as IS and IS+, shows significant differences with a medium and large effect size, respectively. We suspect that even though the similar results between RC and IS in terms of dependability have different origins, i.e., RC is more susceptible to noise while IS does not follow the user's intention as closely, the respective shortcomings have a comparable influence on dependability. The significant differences when comparing IS+ to both other techniques indicate that the extensions made for IS+ provide users with an interaction technique that feels more dependable when interacting with the virtual environment. Compared to RC, the technique is less susceptible to noise, which can increase the feeling of control and security, while compared to IS, the incorporation of the whole object shape for the scoring makes interactions more predictable and thus increases dependability.

Hedonic Qualities When looking at the hedonic qualities of novelty and stimulation, the UEQ results show the same ordering of RC < IS < IS+ in both categories. RC and IS, as well as RC and IS+, show significant pairwise differences with large effect sizes in both categories, indicating that compared to a standard raycasting technique, both techniques are perceived as more novel and more stimulating. This, however, is not surprising as RC is a well-known and straightforward technique and does not provide any interesting modulation to the user's input and, therefore, provides a rather low baseline. When comparing IS and IS+, only a small yet significant effect size can be observed with regard to stimulation, while no significant difference can be observed for novelty. This is not surprising, however, as IntenSelect+ aimed at alleviating shortcomings of IS and not reinventing the technique.

As the motivation was to improve the overall usability of IntenSelect, both categories are important to consider, as gains in objective measurements only do not automatically make the enhancements worthwhile. As stated by Argelaguet et al. in their 2013 survey paper, "In the context of the real usage of 3D interfaces, the subjective impressions of the users about an interaction technique can play a larger role than merely speed" [3]. It can be observed that both pragmatic as well as hedonic qualities show the pattern of RC < IS < IS+. This leads to a large effect size that can be observed between all pairwise comparisons of the techniques, with IS+ being rated as the most attractive, followed by IS and RC. Overall, we can observe that for all UEQ subscales, IS+ has the best score, leading us to confirm $\mathbf{H_7}$.

5.4.3 Generalizability

While the results clearly show that IS+ performs best in the presented environment, the evaluation of IntenSelect+ in other contexts is a key component of future work. One important aspect is that only three different techniques were compared in the study. The two comparative techniques, namely raycasting and the standard IntenSelect technique, were chosen to compare IntenSelect+ against the de-facto standard technique in most VR applications and against the original approach to investigate if the proposed improvements have a profound impact. During the study design, we considered other techniques to compare against, such as Bubble Ray [27], 3D Bubble Cursor [38], Depth Ray [16,38], Alpha Cursor [45]; however, due to the large corpus of relevant techniques, we decided that such comparisons would be out of scope for this work. Therefore, comparisons to other techniques will be part of future work, which allows for adequate room for nuanced discussion between state-of-the-art techniques. Furthermore, we carefully chose representative object configurations that are in line with literature [36]. However, they do not cover all possible object configurations. For example, we did not test different distances and restricted selections to the user's field of view. Therefore, further investigations are needed to ensure there are no specific object configurations where IntenSelect+ falls short. Lastly, objects are represented by pre-defined primitives, which has two potential limitations. Firstly, the primitives must be chosen by the developer/designer, and secondly, the approximation can be too loose. While using compound shapes allows developers to build tighter representations for complex shapes, such as concave objects, it also increases design effort. Therefore, an automated workflow and more complex primitives will be subject of future work.

6 Conclusion and Future Work

While raycasting is a straightforward approach to object selection that is easy to pick up, IntenSelect was designed to overcome its inherent limitations in terms of precise object selection. However, the original realization of IntenSelect was especially suitable for selecting small spherical objects and showed several limitations, such as the complex and inconsistent scoring behavior. Our IntenSelect+ technique presented in this paper proposes enhancements to address these shortcomings. In a formal user study with 42 participants, we compared IntenSelect+ to IntenSelect and raycasting in different spatial object configurations. Our results confirm a clear hierarchical relation between techniques, in which IntenSelect+ showed several improvements over IntenSelect, which in turn was shown beneficial over raycasting. These benefits manifested in objective measures like time and accuracy and subjective measures like task load and user experience. Concerning our research question formulated in the introduction, we conclude that IntenSelect+ is a promising enhancement of the original approach suitable to achieve precise, fast, and comfortable object selections in a large variety of use cases. However, a special case that requires further deliberation for IntenSelect+ is the selection of small objects in front of larger objects, where our study scenario with identical snappiness and stickiness parameters for all objects did not yet lead to optimal selection performance. While our implementation already allows adjusting these parameters on a per-object level, insights on appropriate value ranges and their effects are still subject to further investigation.

Future work will start by exploring variations of IntenSelect+ that are tailored to the needs of specific use cases. Therefore, we aim to add other useful primitives and more complex scoring functions into the technique to approximate arbitrary shapes as closely as possible while maintaining performance, e.g., through the support of k-DOPs. Further ideas to extend IntenSelect+ also include evaluating the impact of dynamic object parameterization based on contextual importance. For example, objects that are likely to receive selections could dynamically increase their snappiness, making their selection in cluttered environments like graphs easier. Lastly, a formal comparison with other techniques and more diverse virtual environments would lead to insights into how IntenSelect+ positions itself among other known methods. While the results were already convincing compared to raycasting and the standard IntenSelect approach, it will be interesting to see how IntenSelect+ scores overall.

Overall, we believe that the underlying conceptual framework of IntenSelect+ can significantly reduce the complexity of selection processes, helping users perform higher-level tasks more effectively and efficiently while improving the user experience.

Acknowledgments

The authors gratefully acknowledge the German Federal Ministry of Education and Research (BMBF) and the NRW state government for supporting this work/project as part of the NHR funding.

REFERENCES

[1] F. Argelaguet and C. Andujar. Improving 3D Selection in VEs through Expanding Targets and Forced Disocclusion. In *International Symposium on Smart Graphics*, pp. 45–57. Springer, 2008. 1, 2

[2] F. Argelaguet and C. Andujar. Efficient 3D Pointing Selection in Cluttered Virtual Environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, 2009. 2

[3] F. Argelaguet and C. Andujar. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics*, 37(3):121–136, 2013. 2, 9

[4] F. Argelaguet, C. Andujar, and R. Trueba. Overcoming Eye-hand Visibility Mismatch in 3D Pointing Selection. In *Proceedings of VRST 2008*, pp. 43–46, 2008. 2

[5] M. Belgardt, A. Bönsch, J. Delember, A. C. Demiralp, J. Ehret, D. Gilbert, K. Karwacki, M. Krüger, S. Oehrl, S. Pape, F. Qurabi, D. Rupp, T. Römer, L. Schröder, and V. Wolf. RWTH VR Group Unreal Engine Toolkit, Dec. 2023. doi: 10.5281/zenodo.10245747 2

[6] L. Besançon, A. Ynnerman, D. F. Keefe, L. Yu, and T. Isenberg. The State of the Art of Spatial Interfaces for 3D Visualization. In *Computer Graphics Forum*, vol. 40, pp. 293–326. Wiley Online Library, 2021. 2

[7] D. A. Bowman, D. B. Johnson, and L. F. Hodges. Testbed Evaluation of Virtual Environment Interaction Techniques. In *Proceedings of VRST 1999*, pp. 26–33, 1999. 2

[8] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. An Introduction to 3D User Interface Design. *Presence*, 10(1):96–108, 2001. doi: 10.1162/105474601750182342 1

[9] D. A. Bowman, C. Wingrave, J. Campbell, and V. Ly. Using Pinch Gloves(tm) for Both Natural and Abstract Interaction Techniques in Virtual Environments. 2001. 2

[10] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2^{nd} ed., 2013. doi: 10.4324/9780203771587 6

[11] G. de Haan, E. J. Griffith, M. Koutek, and F. H. Post. Hybrid Interfaces in VEs: Intent and Interaction. In *Eurographics Symposium on Virtual Environments*, pp. 109–118, 2006. 3, 4

[12] G. de Haan, M. Koutek, and F. H. Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Eurographics Symposium on Virtual Environments*. The Eurographics Association, 2005. doi: 10.2312/EGVE/IPT_EGVE2005/201-209 1, 2, 3

[13] C. Ericson. *Real-time Collision Detection*. Crc Press, 2004. 4

[14] A. Field. *Discovering Statistics Using IBM SPSS Statistics*. Sage Publications, 4^{th} ed., 2013. 6

[15] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture Based Selection for Immersive Virtual Environments. In *Proceedings of UIST 1996*, pp. 95–96, 1996. 2

[16] T. Grossman and R. Balakrishnan. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of UIST 2006*, pp. 3–12, 2006. 2, 9

[17] D. Han, D. Kim, and I. Cho. PORTAL: Portal Widget for Remote Target Acquisition and Control in Immersive Virtual Environments. In *Proceedings of VRST 2022*, pp. 1–11, 2022. 2

[18] S. G. Hart. NASA-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006. doi: 10.1177/154193120605000909 6

[19] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock and N. Meshkati, eds., *Human Mental Workload*, vol. 52 of *Advances in Psychology*, pp. 139–183. North-Holland, 1988. doi: 10.1016/S0166-4115(08)62386-9 6

[20] S. Huang, Q. Daqing, Y. Jiabin, and T. Huawei. Review of Studies on Target Acquisition in Virtual Reality Based on the Crossing Paradigm. *Virtual Reality & Intelligent Hardware*, 1(3):251–264, 2019. 2

[21] W. Kim and S. Xiong. ViewfinderVR: Configurable Viewfinder for Selection of Distant Objects in VR. *Virtual Reality*, 26(4):1573–1592, 2022. 2

[22] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and Accurate 3D Selection by Progressive Refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–74. IEEE, 2011. 1, 2

[23] B. Laugwitz, T. Held, and M. Schrepp. Construction and Evaluation of a User Experience Questionnaire. In A. Holzinger, ed., *HCI and Usability for Education and Work*, pp. 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 6

[24] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. A. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2017. 2

[25] H. Li and L. Fan. A Flexible Technique to Select Objects via Convolutional Neural Network in VR Space. *Science China Information Sciences*, 63:1–20, 2020. 2

[26] J. Liang and M. Green. JDCAD: A Highly Interactive 3D Modeling System. *Computers & graphics*, 18(4):499–506, 1994. 1, 2

[27] Y. Lu, C. Yu, and Y. Shi. 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)*, pp. 35–43. IEEE, 2020. 2, 3, 9

[28] M. R. Mine. Virtual Environment Interaction Techniques. *UNC Chapel Hill CS Dept*, 1995. 2

[29] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin. Moving Objects in Space: Exploiting Proprioception in Virtual-environment Interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 19–26, 1997. 2

[30] A. G. Moore, J. G. Hatch, S. Kuehl, and R. P. McMahan. VOTE: A ray-casting study of vote-oriented technique enhancements. *International Journal of Human-Computer Studies*, 120:36–48, 2018. 2

[31] M. Ortega. Hook: Heuristics for Selecting 3D Moving Objects in Dense Target Environments. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 119–122. IEEE, 2013. 1, 2

[32] A. Pavlovych and C. Gutwin. Assessing Target Acquisition and Tracking Performance for Complex Moving Targets in the Presence of Latency and Jitter. In *Proceedings of Graphics Interface 2012*, pp. 109–116. 2012. 2

[33] I. Poupyrev and T. Ichikawa. Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques. *Journal of Visual Languages & Computing*, 10(1):19–35, 1999. 2

[34] L. Rebenitsch and C. Owen. Individual Variation in Susceptibility to Cybersickness. In *Proceedings of UIST 2014*, pp. 309–317, 2014. 6

[35] M. Schrepp, A. Hinderks, and J. Thomaschewski. Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios. In A. Marcus, ed., *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, pp. 383–392. Springer International Publishing, Cham, 2014. 8

[36] A. Steed. Towards a General Model for Selection in Virtual Environments. In *3D User Interfaces (3DUI'06)*, pp. 103–110. IEEE, 2006. 1, 2, 5, 9

[37] F. Steinicke, T. Ropinski, and K. Hinrichs. Object Selection in Virtual Environments Using an Improved Virtual Pointer Metaphor. In *Computer Vision and Graphics: International Conference, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings*, pp. 320–326. Springer, 2006. 2

[38] L. Vanacken, T. Grossman, and K. Coninx. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2007. 2, 9

[39] Y. Wang and R. Kopper. Efficient and Accurate Object 3D Selection With Eye Tracking-based Progressive Refinement. *Frontiers in Virtual Reality*, 2:607165, 2021. 2

[40] M. Weise, R. Zender, and U. Lucke. How Can I Grab That? Solving Issues of Interaction in VR by Choosing Suitable Selection and Manipulation Techniques. *i-com*, 19(2):67–85, 2020. 3

[41] R. Weller, W. Wegele, C. Schröder, and G. Zachmann. Lenselect: Object Selection in Virtual Environments by Dynamic Object Scaling. *Frontiers in Virtual Reality*, 2:684677, 2021. 1, 2, 3

[42] D. Wolf, J. Gugenheimer, M. Combosch, and E. Rukzio. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–10, 2020. 2

[43] H. P. Wyss, R. Blach, and M. Bues. iSith-Intersection-based Spatial Interaction for two Hands. In *2006 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 59–61. IEEE, 2006. 2

[44] W. Xu, X. Meng, K. Yu, S. Sarcar, and H.-N. Liang. Evaluation of Text Selection Techniques in Virtual Reality Head-mounted Displays. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 131–140. IEEE, 2022. 2

[45] D. Yu, Q. Zhou, J. Newn, T. Dingler, E. Velloso, and J. Goncalves. Fully-occluded Target Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3402–3413, 2020. 9

[46] L. Zhao, T. Isenberg, F. Xie, H.-N. Liang, and L. Yu. MeTACAST: Target-and Context-aware Spatial Selection in VR. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2