frontiers
in Virtual Reality

ORIGINAL RESEARCH
published: 21 June 2021
doi: 10.3389/frvir.2021.684677

LenSelect: Object Selection in Virtual Environments by Dynamic Object Scaling

*Rene Weller, Waldemar Wegele, Christoph Schröder and Gabriel Zachmann**

Computer Graphics and Virtual Reality Research Lab, University of Bremen, Bremen, Germany

Abstract

We present a novel selection technique for VR called *LenSelect*. The main idea is to decrease the Index of Difficulty (ID) according to Fitts' Law by dynamically increasing the size of the potentially selectable objects. This facilitates the selection process especially in cases of small, distant or partly occluded objects, but also for moving targets. In order to evaluate our method, we have defined a set of test scenarios that covers a broad range of use cases, in contrast to often used simpler scenes. Our test scenarios include practically relevant scenarios with realistic objects but also synthetic scenes, all of which are available for download. We have evaluated our method in a user study and compared the results to two state-of-the-art selection techniques and the standard ray-based selection. Our results show that LenSelect performs similar to the fastest method, which is ray-based selection, while significantly reducing the error rate by 44%.

Keywords: virtual reality, 3D interaction, object selection techniques, human computer interaction, interaction design, user interface

OPEN ACCESS

Edited by:
Dirk Reiners,
University of Central Florida,
United States

Reviewed by:
Carlos Andújar,
Universitat Politecnica de Catalunya,
Spain
Dominique Bechmann,
Université de Strasbourg, France

****Correspondence:***
Gabriel Zachmann
zach@informatik.uni-bremen.de

Specialty section:
This article was submitted to Technologies for VR, a section of the journal Frontiers in Virtual Reality

Received: *23 March 2021*
Accepted: *17 May 2021*
Published: *21 June 2021*

Citation:
Weller R, Wegele W, Schröder C and Zachmann G (2021) LenSelect: Object Selection in Virtual Environments by Dynamic Object Scaling. Front. Virtual Real. 2:684677. doi: 10.3389/frvir.2021.684677

1 INTRODUCTION

Selection (a.k.a. target acquisition) of virtual objects in 3D is one of the four universal interaction tasks identified by Bowman et al. (2004). Arguably, a well designed selection technique is one of the most important design aspects of VR experiences (Mine, 1995). This should take into account intuitiveness and seamless usage of the technique, the average number of errors and users' potential frustration, and potential fatigue, all of which can lead to loss of productivity or user acceptance.

In contrast to 2D environments, selection in 3D is more difficult. From a user interface perspective, one of the reasons is the larger number of degrees of freedom (DOFs) inherent in the task. Thus, not only does the user have to control more DOFs, but the mapping between control space and display space can be more complex in the case of non-isomorphic techniques. In addition, lack of precision and fatigue can be an issue, in particular in cluttered environments or with moving targets.

While there are many fundamentally different ways to select objects in virtual environments [see, for instance, Steed (2006) for a taxonomy and Teather and Stuerzlinger (2013), Bacim et al. (2013) as examples], we focus our discussion on selection paradigms that utilize some kind of a 6 DOF pointing device and that use some kind of a single-phase, ray-based paradigm. Reasons are that current consumer VR hardware includes such devices and that ray-based selection techniques are considered very intuitive; also, they do not break the feeling of presence. Moreover, there is evidence that ray-based metaphors yield better user performance than, for instance, image-plane techniques (Bowman et al., 1999).

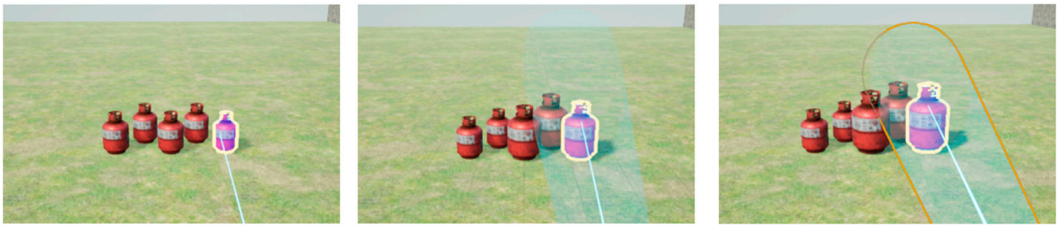


FIGURE 1 | LenSelect in action **(middle and right)** compared to Ray Selection **(left)**.

However, one challenge of ray-based selection techniques in VR is that the effective target width of 3D objects decreases as the distance from the user increases, similar to the effect of perspective foreshortening. This is because the target width as well as its "distance"—which are the essential parameters in Fitts' law (MacKenzie, 1992; MacKenzie and Buxton, 1992; Kopper et al., 2010)—have to be measured in angular space, since users usually keep the position of the input device in place. Thus, the index of difficulty does not only depend on the rotational distance and width in motor space, but also on the object's depth in display space. Of course, this is not only an issue for distant objects, but also for small objects, heavily cluttered scenes with occluded objects, and for moving objects.

We present a novel ray-based selection method called *LenSelect* with a focus on such difficult scenarios. A comparison between our technique and standard ray-based selection can be seen in **Figure 1**. The main idea of our new ray-based selection method LenSelect is inspired by focus + context methods. In order to achieve a seamless and intuitive integration into the virtual environment, we define a non-linear transformation of the geometry that looks to the user like a lens distortion. This could be easily implemented in a shader on the GPU such that only very small changes in the architecture of the VR system are required. Using this lens effect, LenSelect circumvents Fitts' law by increasing the effective target width without affecting the user's spatial presence too much.

In order to evaluate our method, we recognized that there is a lack of standardized evaluation scenarios that would allow to compare the wide variety of selection techniques presented so far. Research has shown that there is no superior selection technique for all situations. Requirements on the task can vary greatly and the conditions for a selection technique can change depending on the layout of the environment, the specific task, and preferences of the user (Wingrave et al. 2005a; Wingrave et al., 2005b; Cashion et al., 2012).

Consequently, we have defined a set of interesting scenarios to investigate ray-based selection methods with a special focus on realistic, dense, and dynamic environments. This comprises the definition of nine test scenes, including a realistic scene with a cluttered distribution of objects in several sizes, but also several artificial scenes, and sets of dynamically moving objects. We have created our test scenes in a widely used, open source game engine, the Unreal Engine, and we will make them available online at blindedforreview, so that other researchers can use them in order to compare more selection techniques. Of course, extraction or re-implementation of the scenes in other VR systems is straightforward, too.

Using our test scenes, we performed a quantitative and qualitative comparison of several state-of-the-art ray-based selection methods in a user study considering typical parameters such as task completion time, selection errors, and Index of Difficulty. In our user study, we included a traditional raycast implementation (referred to as RaySelection henceforth), IntenSelect by de Haan et al. (2005), Expand by Cashion et al. (2012), as well as two different variants of LenSelect. We have included quantitative measures like completion time and error rates but also measure the user experience by using an appropriate questionnaire based on QUESI (Hurtienne and Naumann, 2010).

The results show that RaySelection and LenSelect are the fastest and most intuitive methods in most scenarios; however, LenSelect has significantly less selection errors, similar to Expand, particularly with challenging scenes. There is only one test scene, a scene with fast moving objects, where IntenSelect performs best, but at the cost of significantly worse performance in many others.

2 RELATED WORK

The problem of target acquisition ("selection") in both immersive and non-immersive interactive graphical systems has arisen since the beginning of virtual environments, leading to a huge body of literature. Bowman et al. (2004) presented a taxonomy and a treatize of a number of well-known techniques. In addition, Argelaguet et al. (2008) provide a summary of the classical ray/cone based techniques and an extension thereof.

The ray is a very intuitive metaphor for 3D selection. Consequently, many researchers have looked at methods to improve precision in cluttered environments, in large working volumes, or for hybrid interfaces. One idea is to increase the width of the pointer (Zhai et al., 1994), thus increasing the effective target width (Kabbash and Buxton, 1995). However, more sophisticated techniques are needed to disambiguate the selection, so a number of methods have been proposed, e.g. (Wingrave et al., 2001; Olwal et al., 2003; de Haan et al., 2005; Steed, 2006). Another approach is to change the control-display (C/D) ratio, thus changing the effective target width (Andujar and Argelaguet, 2006; Andujar and Argelaguet, 2007; de Haan et al., 2006). IntenSelect de Haan et al. (2005) relies on a cone-based technique with a disambiguation mechanism based on a scoring function depending on the objects' positions inside the cone. Since it is not always obvious to the user exactly how the disambiguation works, appropriate feedback to the user should be given during the selection. One common practice is to use bent rays (de Haan et al., 2005; Olwal and Feiner, 2003; Riege et al., 2006; Andujar and Argelaguet, 2006; Andujar and Argelaguet, 2007; de Haan et al., 2006).

Another approach is to decompose space, conceptually, into Voronoi regions, with the objects providing the sites. Thus, the regions determine the effective targets widths, which has been proposed with the Bubble Cursor technique (Vanacken et al., 2007). Lu et al. (2020) presented an improved Bubble Cursor technique for selection in 3D environments, with a different definition of the target closest to the ray and different user feedback (including a disk visualizing the bubble and a bent ray). This technique proved to be competitive compared to the older version and seems to be preferred by users (Lu et al., 2020). Our method is different to the Bubble Cursor in that it dynamically increases the objects' actual width; the actual target in Bubble cursor does not have the shape of the object but that of a static convex polyhedron (i.e., the Voronoi region).

Other approaches are to decompose the task, either by dimensions (Pierce et al., 1997; Wyss et al., 2006; Benko and Feiner, 2007), or by successively reducing the set of feasible objects as proposed, e.g., with the SQUAD technique (Kopper et al., 2011). Similarly, Expand (Cashion et al., 2012) improves upon the iterative SQUAD selection technique. With these approaches, objects are often taken out of their original scene context at some point and presented to the user differently (e.g., on an extra grid layer), which could potentially break the feeling of presence.

A few researchers have investigated the efficacy of target expansion during target acquisition (Sarkar and Brown, 1992; Zhai et al., 2003; Hornbæk and Hertzum, 2007; Shoemaker and Gutwin, 2007). The Go-Go technique (Bowman and Hodges, 1997a), and its extension Stretch Go-Go (Bowman and Hodges, 1997a), try to bring objects closer to the user by non-linearly scaling the radial distance of the user's virtual hand from the user's body, while still maintaining the isomorphic scaling of the user's hand in the tangential directions. PRECIOUS works the other way round, teleporting the user close to the target for target acquisition, then sending them back to their initial location (Mendes et al., 2017).

In 2008 Argelaguet and Andujar (2008) presented a similar idea to ours. However, they mainly concentrated on a similar approach to what we call "Same-Screen-Size" scaling and they do not investigate other types of lenses. We will detail the differences to our approach in **Section 3.3**. In contrast to our approach, the authors used a CPU implementation and they had to maintain a graph data structure during runtime which further increases the run time. Moreover, they tested their approach with a limited number of non-standardized test scenes and they did not include dynamic scenes. This makes the results hard to compare. Especially, because their experiments were performed in a Cave instead of HMDs, which are much more common today. Finally, the authors did not include the measurement of the *index of difficulty* for comparison. Actually, the occlusion avoidance they proposed could result in an increasing index of difficulty. However, the results were similar to our approach; both methods have a similar selection time compared to RaySelection but they both reduce selection errors.

Our approach could be seen as an evolution of that idea. In this paper we test multiple approaches as to how to scale the object in the form of four scaling functions.

Other researchers combined different techniques, thus trying to combine the advantages of each to gain additional performance. Techniques can be combined "one after the other" (Grossman and Balakrishnan, 2006), or by using one main technique while the other is acting as a refinement for certain conditions (de Araujo e Silva, 2015).

Most selection methods are computed on the CPU using traditional ray-scene intersection acceleration techniques known from ray-tracing; on the GPU, standard techniques involve usually an extra pass over the scene using a (very thin) frustum defined by the ray. A more recent approach proposes a number of passes implementing successive filtering on the objects and triangles by several occlusion queries (Zhao et al., 2009).

In the area of 3DUI, several techniques have been proposed that are somewhat related to ours. One such technique proposes to manage occlusions in dense 3D scenes by shifting and moving objects "out of the way", where the center of focus is determined by a 3D cursor, hence the name BalloonProbe (Elmqvist and Tudoreanu, 2007; Elmqvist and Tudoreanu, 2006). This technique is different from ours because it does not scale objects in place and the authors have not applied it to perform 3D selection tasks.

3 OVERVIEW OF LENSELECT

In this section we present the design, implementation, and theoretical considerations behind LenSelect in detail. Our method's fundamental idea is rooted in the observation of Fitts' Law that a bigger target width leads to lower selection times and fewer selection errors (MacKenzie, 2018).

The basic formulation of Fitts' Law provides a model for the average selection time MT:

$$MT = a + b \cdot ID \tag{1}$$

Here a, the intercept, and b, the slope, are empirically determined constants, depending on many factors, for instance the type of device. Following this formulation, the Index of Difficulty, ID, can be defined as:

$$ID = \log_2\left(\frac{A}{W} + 1\right) \tag{2}$$

The ID provides a simple measure of the difficulty of a selection task (MacKenzie, 2013). It depends on the distance, A, between the pointer and the object, as well as the target's width, W, in the movement direction of the pointer. Fitts describes W also as the *movement tolerance*, i.e., the error the user is allowed to make while still being able to select the object (MacKenzie, 2018). LenSelect increases this movement tolerance by enlarging the object's effective width in an intuitive way. While the original formulation was derived for one-dimensional target acquisition by Fitts, it is commonly used for 2D and 3D target acquisition, too. In 3D, the target width is usually defined as the angle subtended by an object seen from the user's viewpoint (Kopper et al., 2010).

LenSelect is an extension of the standard ray selection technique. In ray selection, a ray is emitted from the pointing device along the pointing direction. The first object that intersects with the ray is selected when the user triggers the selection button. Ray selection is a very intuitive technique, but can be quite difficult in cluttered scenes or with targets at a distance. The main idea of LenSelect is to modify the *effective size* of the potential target objects thus, consequently, reducing the ID according to Fitts' law. This is done by including kind of a *lens effect* that magnifies objects within a cone, the tip of which is at the pointing device, and the cone's axis is given by the pointing ray.

In general, there are two main parameters in LenSelect: the opening angle of the cone, α, that defines which objects in the scene are affected by the lens effect and, second, the equation by which the final scale is derived. In theory the form of the lens could also be experimented with, however a cone quickly proofed to be the most efficient.

An actual optical lens would scale the whole scene inside it, LenSelect on the other hand scales only individual objects with respect to the distance to the ray. This means we can further reduce overlap between objects, by selectively scaling them. In order to make the scaling and the transitions smooth, we magnify objects the more the closer they are to the center axis of the cone, i.e., the actual ray. Also, those objects are more likely to be selected by the user. To do so, we simply compute the minimum distance, d, between each object's surface (its mesh) and the cone's axis, i.e., we calculate the point on the object's surface closest to the axis and then take that distance.

In the following, we further normalize the distance so that a point on the ray (i.e., cone axis) has distance zero and a point on the edge of the lens has distance one:

$$d_n = \frac{d}{r} \tag{3}$$

where d is the minimal distance between emitted ray and the object's surface and r is the radius of the cone at that distance (see **Figure 2**). This facilitates the definition and presentation of actual scaling functions. In the following, we propose a range of scaling functions, but many more could be defined. For our user study, we selected those two methods performing best in a pre-study. Additionally two hyper-parameters were also evaluated: visibility of the lens and opening angle. According to our pre-study the lens should stay visible (we chose an opacity of 0.1 for increased visibility of the objects) and the opening angle of the lens is 15°.

3.1 Linear Scaling

The simplest method is to scale the object's size linearly with respect to the distance to the cone's border d_n:

$$s = 1 + (1 - d_n) \cdot (s_m - 1) \tag{4}$$

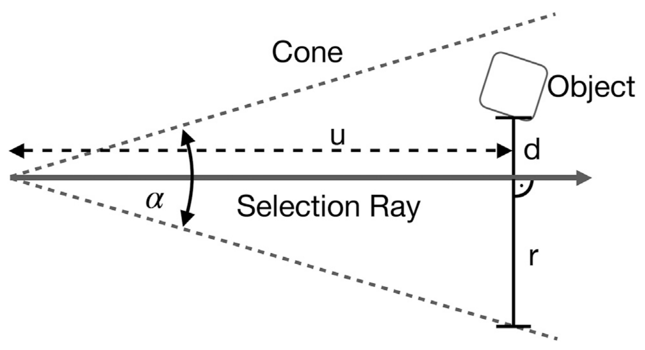


FIGURE 2 | LenSelect scales objects depending on their distance d to the selection ray. Distance d is normalized by the cone's radius r at distance u from the pointing device.

The final object scaling factor, s, depends linearly on a user-defined maximum scaling factor $s_m > 1$. Where $1 - d_n$ describes how much of s_m is applied to the final scale.

3.2 Root Scaling

A disadvantage of the simple linear scaling of the whole object is that it may lead to interpenetrations and occlusions of the scaled objects. In order to reduce this unwanted effect, we propose a method to reduce the scaling factor of objects that are closer to the border of the selection cone. This can be easily achieved by choosing a non-linear scaling factor with respect to the distance. Actually, we decided to choose the root-function that worked well in our pre-tests.

The idea of *Root Scaling* is to emphasize the scaling of objects that are closer to the center of the lens. Again, like with Linear Scaling, we increase the original object's scale up to a maximum scaling factor s_m.

$$s = 1 + \left(1 - \sqrt[4]{d_n}\right) \cdot (s_m - 1) \tag{5}$$

We tested several root functions and chose the fourth root, as the scaling falls rapidly for the first 10% of the distance, but still keeps objects at a reasonable size if they fall outside the initial 10% interval.

3.3 Same-Screen-Size Scaling

Perspective projection has the property that objects appear smaller with an increasing distance to the viewer. If we do not consider this property in LenSelect, it becomes harder to select objects with an increasing distance. When using orthographic projection, on the other hand, objects maintain their size independent of the distance, hence, we could simply use this kind of projection as a lens type. However, perspective projection typically looks better, especially when using stereoscopic displays, because it resembles our normal viewing perception. Consequently, we define a lens type that combines the distance independence of the orthographic projection with properties of perspective projection. Obviously, like the previous lens types, it affects only objects inside the selection cone of LenSelect.

The main idea is to define a scaling factor in world-space of the affected objects in the cone with respect to the distance. For this purpose the *focus depth* n is introduced. Objects that have a distance of n to the camera keep their original scale. From there we need to consider the ratio between the extents of the view frustum at the focus depth f_n and the extent of the frustum at the distance of the object to the camera f_u.

$$s_p = \frac{f_u}{f_n} \tag{6}$$

This way objects that are further away than n will be scaled up, while objects that are closer than n will be scaled down. The latter might be desirable for reducing overlap in dense scenes that are close to the user. The actual extents f_u, f_n can be trivially calculated through simple trigonometry. See also **Figure 3** for a visual overview of the procedure.

In addition, we introduce a nonlinear factor such that smaller objects benefit more from the Same-Screen-Size Scaling, i.e., they tend to appear bigger on the screen than objects that are already large. To do so, we introduce the additional factor $1 + \frac{1}{e}$, where e is the extent of the object in world space. As a simple approximation, we use the average of the three extents of its bounding box (OBB).

Overall the final scale for this lens type is:

$$s_s = \frac{f_u}{f_n}\left(1 + \frac{1}{e}\right) \tag{7}$$

In our pre-study, however, Same-Screen-Size Scaling performed worse than simple Root Scaling. The reason is that objects disruptively “popped” to a different size as soon as they enter LenSelect’s cone. Also, once inside, they never change size irrespective of the pointing direction. This turned out to be a distraction for users in our pre-study. Hence, we did not include it directly in our user study reported below.

Additionally this equation shows some similarity to the scaling factor proposed by Argelaguet and Andujar (2008), but there are differences. Firstly, their equation will never reduce the objects size, while ours does. This has advantages in cluttered environments because the reduced size of objects that are away from the selection ray will also reduce overlap and therefor facilitate object selection. Secondly, in case of a long, narrow object, this object would need to grow into incredible size and therefore create high visual disruption and overlap to satisfy the condition presented by Argelaguet and Andujar (2008). With our equation the scaling factor is only dependent on the distance of the object in relation to a target distance; objects with the same distance to the user will receive the same scaling (if we ignore the additional scaling applied due to target size).

3.4 Combined Scaling

While Same-Screen-Size Scaling assures that far-away objects are not “dominated” by near objects, it introduces a discontinuity in that objects. However, this lens type proofed to be especially useful for the selection of far away objects. In order to maintain this property, we simply combined it with Root Scaling.

This leads to a total scaling factor of:

$$s = 1 + \left(1 - \sqrt[4]{d_n}\right)(s_s - 1) \tag{8}$$

the equation is analogous to 5. Only the constant maximum factor s_m is replaced by the distance dependent scaling factor s_s, retrieved from Same-Screen-Size Scaling. Thus, the minimum scale of an object is 1 close to the border of the cone, while the maximum scale is s_s in the center of the cone.

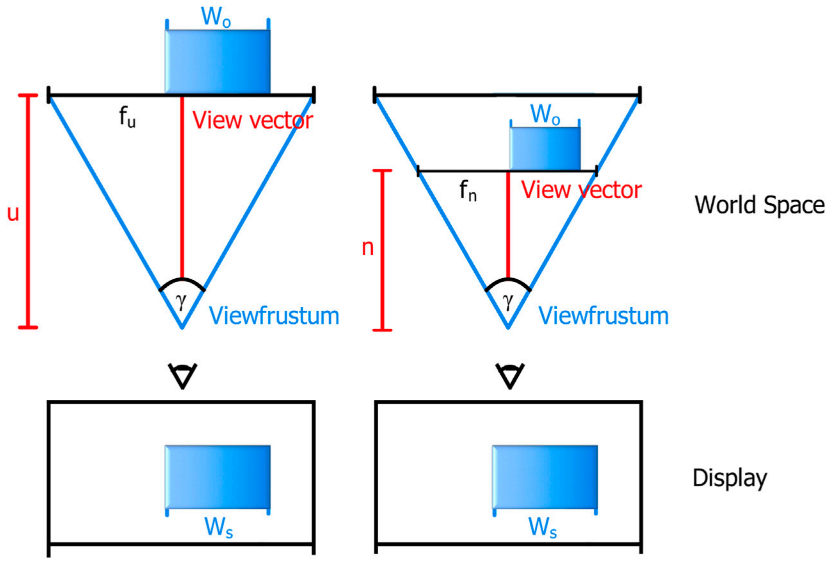


FIGURE 3 | Same-Screen-Size Scaling adjusts the object’s size in world space W_o so that the projection on the screen has the same width W_s regardless of the distance u to the camera. It uses the ratio between the frustums width at distance u and the frustums width at distance n to decide the objects scale. Where n is a user defined value, that describes the distance at which the object keeps its original scale. The left picture shows an enlarged object in comparison to the original scale the object would have at distance n in the right picture.

4 METHODOLOGY

In the section, we describe our evaluation methodology, including the test scenes we have chosen, our method to calculate the ID, and the questionnaire.

4.1 Evaluation of Selection Techniques

Research in selection techniques in the field of human-computer interaction has a long history and there already have been some efforts to standardize test scenarios. For instance, Bowman et al. (1999) propose testbeds to evaluate selection techniques based on taxonomies, performance metrics, and outside factors. Poupyrev et al. (1997) identify a list of different factors, called *task parameters*, that should be considered in the measurement of a selection technique's performance. These parameters can depend on the user, the devices, the interaction technique, the application, and even the task.

However, to our knowledge, there are no standard test scenarios publicly available for 6-DOF selection methods in virtual environments, especially when complex, dense, or realistic scenes should be considered. Often, only simple objects like cubes and spheres are used for the selection. Sometimes, more complex objects are used to simulate an actual use case or a more natural environment, like fruits in a fruit stand.

We propose a new evaluation test bed, which we will make publicly available to the research community. It consists of a set of static and dynamic test scenes with well defined parameters, some resembling rather realistic application scenarios, some rather artificial testing specific characteristics. Our test scenes are best evaluated with our accompanying questionnaire that is based on QUESI (Hurtienne and Naumann, 2010). The questionnaire used (reduced to one selection technique) can be found in the **Supplementary Appendix**??. Finally, we propose methods to calculate width and distance of targets more accurately, especially in dynamic and cluttered scenes, so that the ID in Fitts' law can be computed more accurately.

4.2 Test Scenes

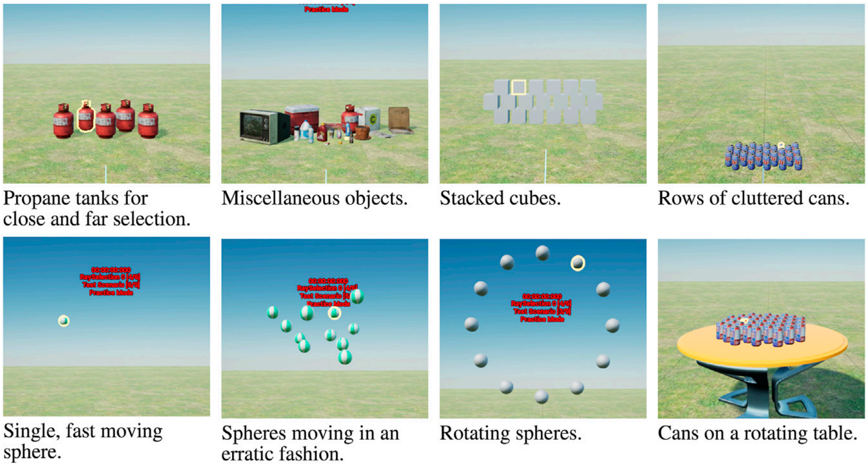
The most challenging selection tasks, especially in real world applications, are cluttered environments. Hence, we focused with high priority on these kinds of scenes. Other challenges that are often not mentioned in test cases are dynamically moving objects. We have designed our test scenes by considering the most important factors proposed by Poupyrev et al. (1997). The test scenarios used by Cashion et al. (2012) served as a basis for our design, but were reworked and adjusted to better conform to the task parameters in set forth Poupyrev et al. (1997). For example, moving cubes were changed to spheres, as the ID remains more consistent that way. We also felt the fruit textures could be distracting, or skew the results in some way, and so chose to eliminate such potentially biasing factors. Each scene was designed to test a specific condition, we felt a high variance in object shapes might have obfuscated valid results. That's why most scenes contain the same objects, with the exception of the *Miscellaneous* scene, which was designed to test performance with objects of differing shapes and sizes specifically.

In order to keep the number of selection tasks manageable for the participants of our user study, we limited the number of test scenes to nine. Most of our scenes were designed to test different aspects of the performance of selection techniques, such as, for instance, different amounts of occlusion or movement in the scene. The "Miscellaneous" test scene bears concrete resemblance to a real world scenario, in an effort to include evaluation "in the wild". Overall, we think our test scenes cover a wide variety of interesting and practically relevant cases.

We hope to provide a set of scenarios that is suitable for future evaluations of selection techniques to come, since comparing different selection techniques can be difficult with the currently wildly differing design of selection tasks. The following lists our test scenes, including our rationale for their design; images of all scenarios can be found in **Figure 4**.

- Propane Tanks close: this scenario evaluates the influence of the user's distance as the main influence factor in a realistic scenario. It consists of two rows of propane tanks with only little occlusion; three tanks are in the front row and two in the back. We test this scenario for two different distances.
- Propane Tanks far: is exactly like Propane Tanks close, except at a far distance.
- Miscellaneous: this evaluates the influence of the object's size on the selection technique. It consists of different realistic objects of varying sizes and shapes that are slightly cluttered.
- Stacked Cubes: this scenario is closely related to traditional 2D selection tasks and thus, evaluates the performance for such easy, almost two-dimensional tasks. A number of cubes are stacked in three rows in front of participants.
- Densely Placed Cans: this scenario evaluates the influence of heavy occlusion on selection performance; three rows of cans are closely placed behind each other.
- Fast Moving Single Sphere: in order to evaluate the performance on fast moving objects with unpredictable trajectories, we include this scene where a single sphere moves around randomly in the scene.
- Erratically Moving Spheres: this scenario covers the test cases with unpredictable moving objects with slight occlusion: ten spheres move around in an erratic fashion. They move slowly, but also randomly change direction.
- Floating Spheres: this scenario also tests 2D selection tasks, i.e., it does not have occlusions, but in contrast to Stacked Cubes, it focuses on moving objects with a predictable trajectory. Twelve spheres are placed at a certain distance from the user. They slowly rotate around a center, similar to a wheel.
- Rotating Cans: the ultimate challenge are highly occluded small objects in motion. Hence, like above we placed small cans on a table in close proximity but now, additionally on a rotating table.

In order to establish a well-defined timing for the task completion and a well-defined (angular) distance to the object to be selected, the user has to touch a sphere that sits at the top center of the whole scene before each selection trial. As soon as



Propane tanks for close and far selection.

Miscellaneous objects.

Stacked cubes.

Rows of cluttered cans.

Single, fast moving sphere.

Spheres moving in an erratic fashion.

Rotating spheres.

Cans on a rotating table.

FIGURE 4 | Our nine different test scenarios we propose and that were used in this study (the top left, Propane tanks, stands for two scenes Propane tanks close and Propane tanks far. The top row contains the static scenes, while the bottom row contains that dynamic scenes. Both the static and the dynamic subsets of scenes contain more as well as less cluttered environments.

the user touches this start sphere, the timing for the selection task begins and the target object is highlighted in the scene. This ensures that the recorded time does not include time needed for searching the target object. It also ensures that every participant has a similar distance to cover to reach the target object. Of course, all users were positioned at the same place in the virtual environment and were not allowed to walk closer.

4.3 Index of Difficulty Calculation

Computing the difficulty of a selection task can be generally done by using Fitts' law, as mentioned above. However, computing the distance and, especially, the width of objects can be challenging for complex object shapes, or for occluded, or even moving objects. In the following, we propose a method how to evaluate the ID in such challenging scenarios.

In general, the actual distance and width, needed for the ID, can be calculated by taking a screenshot from the camera's view and counting the pixels from the pointer's original position to its final position during selection (see **Figure 5**). We have implemented our test scenes in the Unreal Engine. The Unreal Engine's custom stencil buffer was used to identify the affiliation of each pixel, which allows to identify objects by their respective identifier stored in the stencil buffer's pixel. In addition, a custom color can be applied when the buffer is retrieved. In our case, occluding objects are colored cyan, the target object is colored yellow, everything else is colored black. Consequently, given the intersection points between the selection ray and the image plane, we can just trace the path between those points and count the pixels; these numbers can be converted into (angular) width and distance and plugged into **Eq 2** in order to determine the ID of each trial. One should keep in mind that only selectable parts of an object are counted as it's width. Holes in objects are not included, refer to **Figure 5**, where red pixels signify the width of the object for a given selection.

For Expand and IntenSelect, we calculated the actual ID's slightly different, this is because the standard approach does not work in these cases.

For Expand the object's width remains the same, while the distance consists of two parts. The first is the distance from the starting position of the pointer to the position where the user decides which objects are of interest to them (which objects will be selectable in the grid). The second part covers the movement from the first rough selection to the selection of the target object in the grid. Basically we consider the actual path of the pointer before it can select the target object.

For IntenSelect the user does not need to directly point at the object, therefore we have to compute the target region. The target region describes all pixels which would lead to selection of the target object. This is done by calculating the cone's radius at the object's depth in world-space and projecting the resulting circle onto a canvas, so that its center is aligned with the projected center of the object. Then the score for each pixel inside this circle is computed and the pixel colored appropriately. After that the ID can be calculated as usual with the techniques described above.

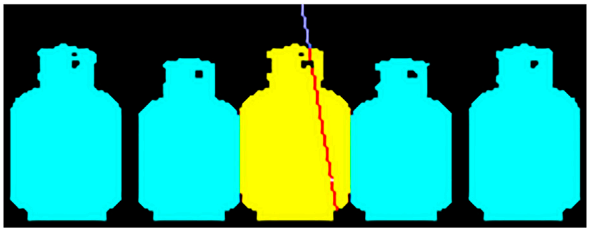


FIGURE 5 | Screenshot visualizing our ID calculation using the stencil buffer. The blue part of the line shows the distance to the target, while the red part(s) show the width of the target. The target object is represented by yellow pixels.

For moving objects, the Index of Difficulty must be defined somewhat differently to account for the target's velocity relative to the user. Even though research into this specific problem is rather old (Jagacinski et al., 1980), resources are scarce. Using a first order control system, Hoffmann (1990) derived an equation close to Fitts' original ID:

$$ID = \log_2\left(\frac{A + \frac{V}{K}}{\frac{W}{2} - \frac{V}{K}}\right) \tag{9}$$

A describes the distance to the target, while W describes the targets width, same as in Equation 1, where V is the velocity of the target, where an approaching target is defined to have negative velocity. K serves to determine the critical speed V_{crit} at which deliberate target acquisition is no longer possible:

$$V_{crit} = \frac{W \cdot K}{2} \tag{10}$$

The value of K is difficult to measure; Hoffman proposed two methods: first, by regression analysis and, second, by observing the critical speed at the "threshold" to loss of target acquisition (50% of unsuccessful trials). Hajri et al. (2011) derived an equation similar to that of Hoffman by modeling target acquisition time.

We chose Hoffman's equation for calculating ID's of moving objects. We (more or less arbitrarily) chose $K = 100$ after some experimentation to derive IDs that were in the same order of magnitude of those of stationary objects.

For the stationary ID calculation, we rely on **Eq 2** that is often used for 2D target acquisition. Several modifications, such as accounting for gravity (Murata and Iwase, 2001; Grossman and Balakrishnan, 2004), where proposed to adapt the calculations to specific scenarios in 3D. Shoemaker et al. (2012) described further limitations with regards to different levels of control-display gain. They propose a two-part formulation for selection techniques with non-isomorphic mappings. All selection techniques used in our study use an isomorphic mapping, hence we do not consider it.

4.4 Questionnaire

Beyond pure quantitative measures like task completion time, error rate, etc, the user experience is essential for judging interaction methods. This is usually measured by a questionnaire. We propose an extension of the QUESI questionnaire to adapt it to the special requirements of selection tasks in 3D environments. Usually, the QUESI questionnaire is used to measure the subjective criteria of intuitive use and user satisfaction (Naumann and Hurtienne, 2010). It connects intuitive use with effective interaction by the user, hence it offers a good basis on the user's acceptance of a selection technique. QUESI consists of 14 questions, each corresponds to one of the following five scales: cognitive load, perceived achievement of objectives, perceived learning effort, familiarity/pre-knowledge, and perceived error rate. The mean value of the corresponding questions, measured typically on a five-point Likert scale, constitutes the score for each scale. The mean of all the scales defines a single total QUESI number, which can be used for overall comparison.

According to Poupyrev et al. (1997) it is essential to maintain the sense of presence during selection tasks in 3D VR environments. Hence, it is useful to include this important measure in the user evaluation. QUESI is missing this scale, but the INTUI questionnaire includes it under the term "Magic Experience" (Ullrich and Diefenbach, 2010). According to INTUI, four items correspond to this scale but one of the questions is very direct and includes the term "Magic Experience". Definition and interpretation of this term is vague, therefore we removed this question and incorporated the other three question into our extended QUESI questionnaire. Consequently, our modified QUESI is based on six scales instead of five. Hence, the direct total QUESI number cannot be directly compared to results in prior work, but we hope that in the future more researchers will rely on our testing method and the corresponding questionnaire.

Additionally, we added two items to measure the complexity and the fun factor of an interaction method by directly asking: "How complex did you find the selection technique?" and "Using the selection technique is fun". All questions rely on a five-point Likert scale.

In our user study described in the next section, we also asked the participants at the end to rank all selection techniques by their preferences, and we included a short pre-questionnaire asking for demographic data like prior experience with VR and video games, especially first-person shooters. Also, we added an open post question asking if a participant had problems with the study itself or the system used to conduct the study.

5 USER STUDY

In this section, we will describe our study design as well as the details of the three selection techniques we compare our work against. In total, we compare five different selection techniques (two of our own) using the nine test scenes described in **Section 4.2**.

5.1 Study Procedure

The study uses a within-subjects design where all 20 participants evaluated all five techniques. We recorded quantitative data, like task completion time, selection errors, and the Index of Difficulty (ID) as defined in **Section 4.3**. For static scenes, we calculate the physical ID in addition to the effective ID. The physical ID is the ID without any scaling applied by the selection method, whereas the effective ID takes the target scaling into account. Additionally, all subjects filled out the questionnaire after using each technique (so, in total, each participant had to fill out five questionnaires).

Participants entered a designated spot in the virtual environment to assure equal distance to the test scenes. For each selection technique, participants first entered a *practice mode* without a time limit. When feeling comfortable, participants entered the actual study. They had to make multiple selections for each scene: 10 when selecting the propane tanks, 20 in the miscellaneous scene, and five in all other environments. Every selection technique was evaluated on all scenes, as described in **Section 4.2**. To avoid learning effects between techniques, we randomized the order of the selection techniques and scenes according to a latin square.

We felt different numbers of trials were appropriate in order to reduce potential learning effects in scenes with a small number of target objects. Otherwise, muscle memory might skew the results of successive repetitions of rounds. This was less of a problem in scenes like the Miscellaneous environment, since here every selection targeted a different object, at a different location, with a different size and form. Another reason was that we did not want to strain participants' patience too much, so as not to induce fatigue or boredom (they were not compensated for their participation).

In the following, we give a short recap of two competing state-of-the-art selection techniques we included in our user study. First, we chose IntenSelect because of it's basic premise is similar to that of LenSelect. Second, Expand was chosen because it is a well-known member of the family of progressive, multi-stage selection techniques, which tend to be slower, but less prone to selection errors. Last, we included the classic ray-based selection, as kind of a baseline.

5.2 Competitors

5.2.1 Expand

Expand is an evolution of the SQUAD selection technique; both are multi-stage approaches to object selection. Expand tries to alleviate the problem of losing the original context of the objects, especially useful with similarly looking objects. Instead of iteratively arranging the candidate objects into screen-space quadrants, Expand aligns copies of the objects in a screen-space grid. This grid is dynamically generated depending on the number of objects that need to be placed such that it maximizes screen space. In the second stage, the user selects the target from the grid (Cashion et al., 2012). See the right hand side of **Figure 6** for a screenshot of this second stage in our implementation.

Expand is not without problems, however. For example, the arrangement of objects in the grid is not ordered according to an intuitive criterion. Also, objects completely occluded in 3D show up in the grid as well, which may or may not be confusing to users.

Expand's main purpose in our study is to serve as a comparison for selection errors. As a two-stage selection technique, the possibility for errors should be lower than the other single-stage techniques, thus providing kind of a baseline with respect to this performance metric. We think it is important to include a representative of the family of multi-stage selection techniques, even though we don't expect it to outperform any of the single-stage techniques in terms of task completion time.

5.2.2 IntenSelect

IntenSelect was designed to tackle three aspects of selection: accuracy, ambiguity, and complexity de Haan et al. (2005). IntenSelect is a cone-based technique with a disambiguation mechanism. It uses a bending ray emanating from the device in the pointing direction and bending toward the currently selected object. In addition, it renders a second, straight ray pointing in the device's direction (but this could be omitted). See the left hand side of **Figure 6** for a screenshot of our implementation.

IntenSelect proposes a scoring mechanism for objects that is based on the angle subtended by the object's center with the cone center line. This angle is given by $\arctan \frac{d_{perp}}{d_{proj}}$, where d_{perp} is the object's distance from the cone's center line, and d_{proj} is the distance along that line. The authors further introduce a parameter k to "narrow" the scoring with depth, such that the final, momentary score is

$$s_{contrib}(t) = 1 - \frac{\arctan \frac{d_{perp}(t)}{\left(d_{proj}(t)\right)^{k}}}{\beta} \qquad (11)$$

where β is a parameter defining the opening angle of the cone. In order to avoid fast-switching behavior between targets, they introduce kind of a temporal hysteresis, thus arriving at the final score

$$s_{total}(t) = s_{total}(t-1) \cdot c_s + s_{contrib}(t) \cdot c_g \qquad (12)$$

where parameters c_s and c_g define "stickiness" and "snappiness", resp. Thus, the effective target width of each object is expanded into a sphere around it's center point, depending on other objects in the vicinity.

However, this method can lead to problems in highly cluttered areas: In such cases, the effective target region of an object can shrink below the size of the target object. This effective target region could even be "shifted" away from the target object visible to the user. In the worst case, the target region can lie completely outside the visible target object; this is the case, if there are close objects in front of and behind the target object. Another unintuitive situation can occur, for example, if the target object's center point is overlapped by another object; then, the target object can still be selected through the overlapping object.

6 RESULTS

We have used the methodology described above to perform a user study. We will first present the general results from our study. Then we focus on the quantitative findings before we report the results from the questionnaire.

The main study had 20 participants, 15 of which self-identified as male and five as female. Most of them were either research assistants or students. Four people were left handed and none had any form of color blindness. Their ages ranged from 20 to 35 with a mean age of 26.7, a standard deviation of 4.3 and a median of 27.5. All but two participants had used a HMD before. A total of seven participants already participated in at least one of the pre-studies.

Before the main study, we conducted two pre-studies with 7 participants each to determine the hyper-parameters for LenSelect. In the first pre-study, we found that the Root and Combined Scaling methods were the most promising versions of LenSelect. From the second pre-study, we found that an opening angle α of 15° was the best option and that visualizing the cone worked better than hiding it.

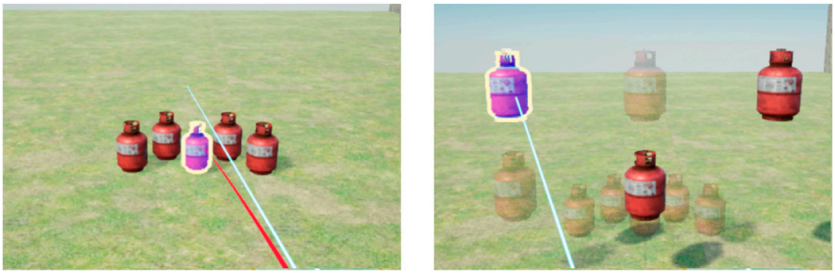


FIGURE 6 | Screenshots of our implementations of two state-of-the-art selection techniques used in our user study: IntenSelect (left) and Expand (right).

In all experiments, the participants wore the HTC Vive Pro. Two base stations were used to track user movement. To indicate selection participants used the trigger button. Although this might induce selection errors through the Heisenberg effect of spatial interaction (Wolf et al., 2020), since all techniques use the same button for indication the error should spread among them equally. The HMD was connected to a Windows 10 PC with an Intel Core i7 7800X at 3.5 GHz, 64 GB RAM and a NVidia Titan V GPU. We implemented all test scenes and selection techniques with the Unreal Engine 4.21.2.

6.1 Statistical Methods

We use analysis of variance (ANOVA) for a statistical interpretation of the following results. Both normality testing with the Shapiro-Wilk test and visualizing the data as a histogram shows that samples do not necessarily follow a normal distribution for both the questionnaire and quantitative data. The Levene test also shows that homoscedasticity does not always apply to the quantitative results. Therefore, we chose Welch's ANOVA with subsequent Games-Howell post-hoc test to analyze the quantitative data. The Welch test is recommended if the data is not normally distributed, and the variances are unequal. The test is robust regarding differing sample sizes and non-normal data, even if some conditions apply (Cribbie et al., 2012). The Games-Howell test is robust to non-normality as well and works with unequal variances.

For the results from the questionnaire, we chose the Kruskal–Wallis test with subsequent Tukey post-hoc test. Parametric and non-parametric tests often perform equally well for Likert scale data with large sample size (>15) (Mircioiu and Atkinson, 2017). We chose the Kruskal–Wallis test because our sample size is often smaller than 15. This test assumes equal variances across the data. As Mircioiu and Atkinson show, for many situations, both homoscedasticity and normality do not play a role, as long as sample sizes are (nearly) equal (Mircioiu and Atkinson, 2017). Both Welch's ANOVA (Honjing et al., 2016) and the Kruskal–Wallis test (Honjing et al., 2016) are somewhat robust to outliers. We use the standard significance level $\alpha = 0.05$, as both of our tests already adjust for multiple comparisons.

6.2 Quantitative Data

We had to remove five selections due to questions of the user or distractions: two for Expand, two for LenSelect Combined, and one for IntenSelect. A small amount, 68 out of the total of 1,500 selections, were removed because their ID could not be calculated. The calculation failed, for example, when the end point of the selection was outside the participant's visible area.

First, we investigated the task completion time. In all scenes we found significant results. The classical RaySelection, both versions of LenSelect and and IntenSelect performed significantly faster than Expand in all scenes. In case of tidy scenes like the Fast Moving Single Sphere or the Rotating Spheres, IntenSelect performs best. For instance, in the Fast Moving Single Sphere the Games-Howell post-hoc test achieved a significance difference ($p < 0.001$) for IntenSelect compared to RaySelection (−0.688s, 95%-CI[−0.802s −0.573s]) and LenSelectRoot (−0.215s, 95%-CI[−0.291s −0.139s]). On the other hand, the different LenSelect versions and RaySelection outperform IntenSelect significantly in highly cluttered scenes like the Cluttered Cans or the Rotating Cans Scenario when it comes to selection times. For instance for the Rotating Cans Scenario we measured a significance of ($p < 0.001$) for LenSelect Combined with the Games-Howell post-hoc test compared to IntenSelect (0.668s, 95%-CI[0.411s 0.925s]). In the severely cluttered scenes, like the Cluttered Cans, RaySelection, LenSelect and IntenSelect performed similarly with respect to the selection time. **Figure 7** shows the results for all scenarios. The interested reader will additionally find all results from our statistical analysis using one-factorial ANOVA tests and Games-Howell post-hoc analysis in the addendum (**Supplementary Table S1**).

Beyond the pure selection time, it is essential that the user really selects the correct object. Hence, we also investigated the error rates for the individual scenarios. **Figure 8** gives an overview on the average errors made with the different selection techniques. In detail, for the slightly cluttered static scenes, we found only very little errors for all selection techniques. The mean error was <0.06, the median was 0, and the standard deviation <0.3 errors. In the tidy dynamic scenes, i.e., the Rotating Spheres and the Fast Spheres, IntenSelect performs significantly better than the other selection techniques. For instance, in the Fast Spheres scenario we measured a significant difference ($p < 0.001$) for IntenSelect compared to RaySelection (−0.86, 95%-CI[−1.00 −0.715]) and LenSelect Combined (−0.32, 95%-CI[−0.420 −0.219]). However, LenSelect Combined also produces significantly ($p < 0.001$) less errors than RaySelection (−0.54, 95%-CI[−0.714 −0.365]). In highly cluttered scenes, either static

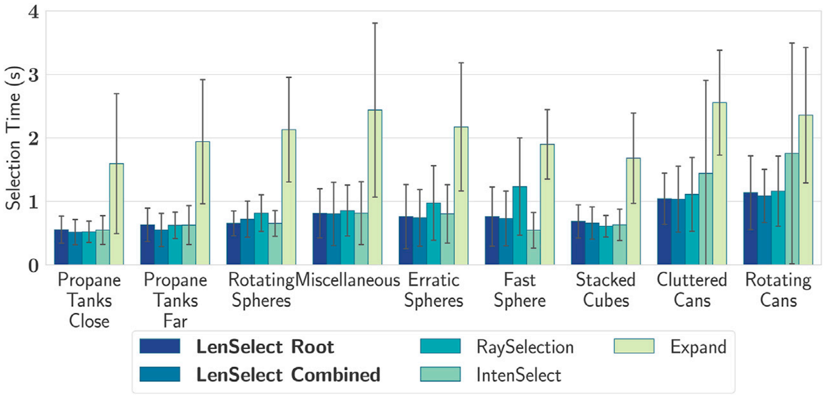


FIGURE 7 | Average time for one selection for each selection technique. As can be seen LenSelect produces better selection times than IntenSelect and slightly better times than RaySelection. Whiskers show the one standard deviation interval.

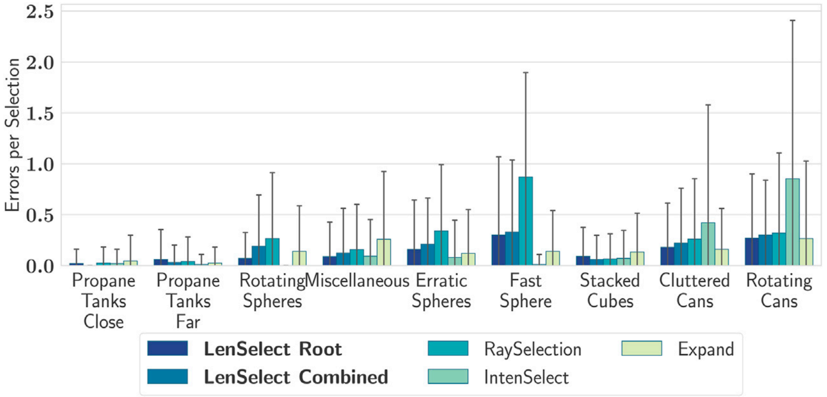


FIGURE 8 | Average errors for each selection technique. The whiskers show the 1 standard deviation interval.

or dynamic like the Cluttered Cans and Rotating Cans, both LenSelect versions outperform the other methods. For instance, in the Rotating Cans scene, LenSelect produces significantly fewer errors ($p < 0.001$) than IntenSelect (0.119, 95%-CI[0.316 0.788]) according to the Games-Howell post-hoc test.

Figure 9 provides an overview of the average effective Index of Difficulty for each selection technique and test scene. Since all selection techniques modify the ID (except RaySelection, obviously), we also investigated the influence of the physical ID on the selection time (see **Figure 10**), where we fit a line into all samples where a physical index of difficulty could be computed. Overall, the LenSelect Combined performs best with respect to the physical index of difficulty, except for very low ID's (i.e., easy targets), where the standard RaySelect performs best.

Questionnaire Data

In the complexity rating from the questionnaire (see **Figure 11**, left), the Kruskal–Wallis test shows significant effect ($H(4) = 30.63, p < 0.001$). The Tukey post-hoc test reveals that Expand ($M = 3.35, SD = 1.14$) is rated significantly worse than all other selection techniques. IntenSelect ($M = 2.45, SD = 1.15$) is rated significantly worse than RaySelection ($M = 1.5, SD = 0.83$). RaySelection was rated the least complex, followed by LenSelect Combined ($M = 1.70, SD = 0.73$) and then LenSelect Root ($M = 1.95, SD = 0.83$).

The rating wrt. hedonistic user experience also shows a significant effect ($H(4) = 14.70, p < 0.05$). Expand ($M = 1.60, SD = 1.31$) performs significantly worse than RaySelection ($M = 3.95, SD = 0.94$), LenSelect Root ($M = 3.80, SD = 0.89$), IntenSelect ($M = 3.70, SD = 1.45$) and LenSelect Combined ($M = 3.95, SD = 0.76$). The others seem to perform equally well compared to each other.

The QUESI Total is significant as well ($H(4) = 29.14, p < 0.05$). RaySelection ($M = 4.41, SD = 0.42$) performs significantly better than IntenSelect ($M = 3.64, SD = 1.0$) and Expand ($M = 3.22, SD = 0.86$). While Expand shows a significantly lower rating than LenSelect Root ($M = 4.25, SD = 0.55$) and LenSelect Combined ($M = 4.22, SD = 0.40$). Both LenSelect techniques are rated slightly lower than RaySelection, but not significantly so.

When the users where asked to rank the selection methods from most to least favorite (see **Figure 11**, right), a significant effect can also be observed in the user ranking ($H(4) = 26.93, p < 0.001$). Here Expand ($M = 1.55, SD = 1.05$) is significantly less popular than

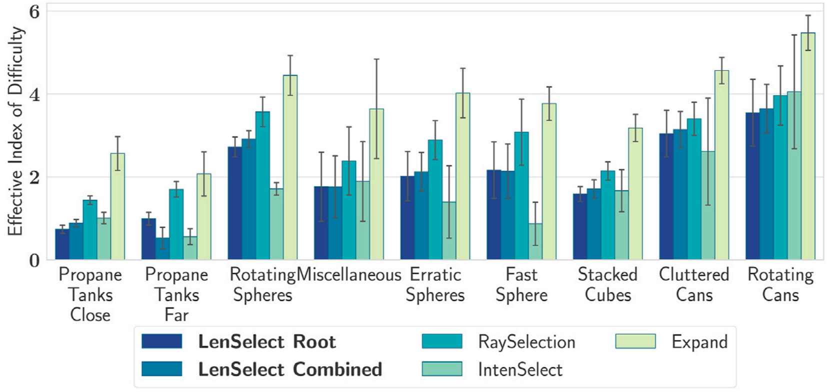


FIGURE 9 | The average effective Index of Difficulty for each selection technique. The whiskers show the 1 standard deviation interval.

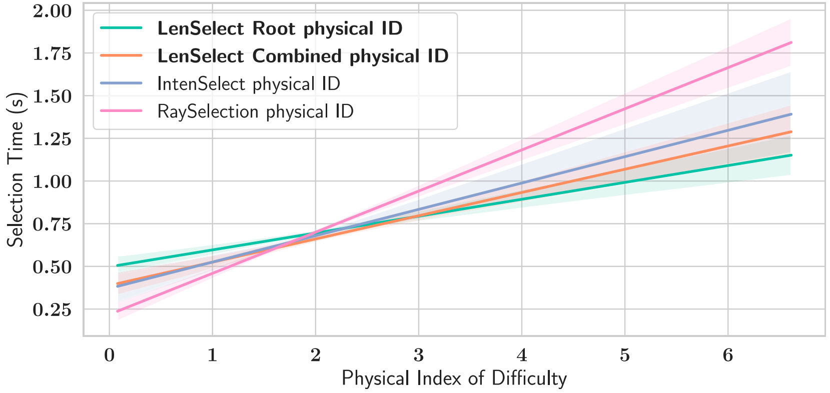


FIGURE 10 | Relationship between task completion time (selection time) and the targets' physical ID, which is the ID the targets have in case of the simple, standard RaySelection. The selection times were averaged over all test scenes and all participants. LenSelect Combined performs better than the other three techniques, except for very low ID's (i.e., very easy targets).

RaySelection ($M = 3.40, SD = 1.27$), LenSelect Root ($M = 3.5, SD = 0.88$), IntenSelect ($M = 3.05, SD = 1.54$) and LenSelect Combined ($M = 3.40, SD = 1.35$). The other techniques show no such differences between each other.

7 DISCUSSION

The results from the study show that no selection technique performs best in all scenes and with all metrics. There are some general trends, though.

According to users' subjective overall satisfaction (QUESI score), RaySelection was rated highest among all selection techniques, which is mostly due to its high ranking regarding complexity (i.e., users felt it is the least complex technique). However, it performs very slow wrt. task completion time, except for the simplest scenarios (i.e., very low physical IDs). Also, RaySelection has relatively high error rates in many scenes, particularly with moving targets.

Expand performed, on the one hand, worse than all other methods wrt. the subjective metrics (i.e., results from the questionnaire) and wrt. the (objective) selection time. On the other hand, it showed to be the least error prone in most cases, which was to be expected. Interestingly, it did not always result in the least amount of selection errors. The reason for that is not obvious to us, so would require further investigation.

IntenSelect worked best if there was only little to moderate occlusion with other objects. In these scenes, this technique can generate the best effective target region for the objects. In other scenes, it led to many selection errors and a high selection time. Further, it was rated worse than both LenSelect and RaySelection in all subjective metrics.

LenSelect performs well more consistently than all other techniques over all test scenarios with respect to the selection time and selection errors (see **Figure 10**). This consistent performance could explain why both LenSelect variants were favored in the user rankings.

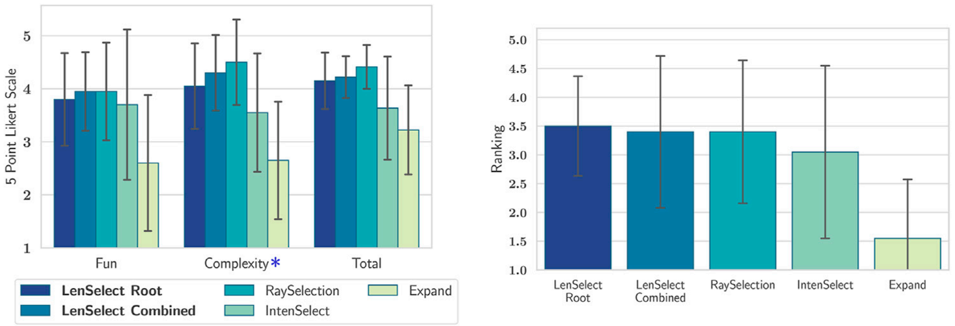


FIGURE 11 | Left: Participant's average feedback for the factors fun, complexity and total score from the QUESI questionnaire. Higher values are better on the 5-point Likert scale. (The results for complexity were inverted, so a higher value here means less complex.) Right: Average user's popularity ranking. The most popular technique got five points, the second most popular got 4 and so froth. Whiskers show the one standard derivation.

It is also of note that there is a difference between LenSelect's variants Root and Combined Scaling in certain scenarios. It appears that LenSelect Combined performs better than LenSelect Root with distant objects, which showed in the scene "Propane Tanks Far". (This was the reason why we selected LenSelect Combined in our final study.) This difference, however, is not as pronounced as it might be, since all other test scenes had about the same distance to the user. At middle to close ranges, both scaling functions for LenSelect perform equally well according to our studies. Most of the differences wrt. task completion time are not significant and even if they are, the effect size is usually so small that it's questionable if the difference is even noticeable to the user.

Lastly, we think that LenSelect has the potential to completely replace RaySelection in pretty much all use-cases. LenSelect performs equally well under circumstances where RaySelection shows good results and otherwise outperforms it in all other cases tested. The main advantage of LenSelect is its consistency, as it never shows any really bad performance in any of our test scenes.

Although all selection techniques are implemented on the CPU in the study the performance overhead for LenSelect was minimal. Using the *Miscellaneous* test scene for comparison both LenSelect techniques were able to comfortably run at 60 fps with no change in performance when moving the cursor and therefor scaling objects. Compared to RaySelection, which also ran at 60fps the performance overhead is minuscule and should be negligible when LenSelect is implemented in a shader.

8 CONCLUSION AND FUTURE WORK

We have presented a novel ray-based selection metaphor called LenSelect that implements kind of a lens distortion effect, thereby increasing the effective target width dynamically. LenSelect is easy to implement and can be integrated in existing VR systems without much changes required. In fact, it can be implemented in a shader, thus requiring minimal to no changes to the rendering code.

Moreover, we have presented a methodology to compare ray-based VR selection methods. Our methodology includes a set of scenes that cover the most important factors influencing selection techniques, i.e., the distance, the size, and the density of the objects in the scene; it also includes an accompanying questionnaire to evaluate user feedback.

Finally, we conducted a user study to compare several state-of-the-art selection techniques. Our results show that LenSelect can compete even with classic ray-selection with respect to selection times while providing a significantly reduced error rate by up to 44%. IntenSelect performs better than LenSelect in test cases with distant objects but has problems in near cluttered dynamic scenes. All single-stage, ray-based selection techniques in our user study, i.e., RaySelection, LenSelect, and IntenSelect, outperform Expand, which uses a two-stage approach.

Overall, LenSelect's good performance seems to be the most consistent among all selection techniques we compared in this study across all of our test scenarios. It also achieved the best acceptance by the users.

The lens-effect-based approach of LenSelect seems generally very promising and, with further adjustments, it might even outperform IntenSelect in scenes with large target distances. To achieve that, LenSelect offers many possibilities to modify the lens effect.

One such modification could be to dynamically change the order of the root in Root Scaling, depending on the number of objects inside the lens (the more objects, the higher the root order). That way, selecting objects from very cluttered environments becomes even easier. Another option is to calculate the scaling factor of Same-Screen-Size Scaling at runtime, but then use that as the maximum factor s_m in Root Scaling which should improve the performance especially for distant objects.

Furthermore, the performance in scenes containing a very high variability of object sizes (especially with tiny objects) could be improved. For instance, a dynamic change of the particular lens effect at run-time, depending on the scenario, could be an option. The general framework of our LenSelect technique allows for easy integration of other or multiple lens distortion effects. Obviously, such a changes should not break the user's presence in the VR environment; for instance, a continuous blending between several lens types can be considered.

In the future, we plan to further investigate other ray-based selection methods. Of course, we also encourage other researchers to perform their experiments with our methodology. In order to do so, we make our scenarios publicly available at blindedforreview.

Moreover LenSelect has only been compared to ray-based selection techniques. In especially cluttered environments other approaches might prove more fruitful, like Starfish (Wonner et al., 2012) or teleportation (Mendes et al., 2017).

Finally, we plan to use our testing methodology to investigate the influence of the input device and the kind of VR display on the optimal selection technique. For instance, we will test more sophisticated input devices for selection tasks such as natural interaction with hand tracking and investigate selection metaphors for large projection powerwalls instead of HMDs.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

RW has contributed to the development of the technique, the statistical analysis, and the writing of the paper. WW has implemented the technique(s), performed the user studies, contributed to the statistical analysis and the writing of the paper. GZ has conceived the initial idea of the interaction technique, contributed to the development of the technique and the writing of the paper. CS contributed to the development of the technique, the statistical analysis, as well as the writing of the paper and provided additional assistance.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frvir.2021.684677/full#supplementary-material

REFERENCES

Andujar, C., and Argelaguet, F. (2006). "Friction Surfaces: Scaled ray-casting Manipulation for Interacting with 2D GUIs," in 12th Eurographics Symp. on Virtual Environments (Lisbon, Portugal: EGVE 200).

Andujar, C., and Argelaguet, F. (2007). "Virtual Pads: Decoupling Motor Space and Visual Space for Flexible Manipulation of 2D Windows within VEs," in IEEE Symp. on 3D User Interfaces (IEEE).

Argelaguet, F., and Andujar, C. (2008). *Improving 3d Selection in Ves through Expanding Targets and Forced Disocclusionsmart Graphics*. Berlin, Heidelberg: Springer).

Argelaguet, F., Andujar, C., and Trueba, R. (2008). "Overcoming Eye-Hand Visibility Mismatch in 3d Pointing Selection," in Proc. 2008 ACM Symp. on Virtual reality software and technology (New York, NY, USA: ACM), 43–46. doi:10.1145/1450579.1450588

Bacim, F., Kopper, R., and Bowman, D. A. (2013). Design and Evaluation of 3d Selection Techniques Based on Progressive Refinement. *Int. J. Human-Computer Stud.* 71, 785–802. doi:10.1016/j.ijhcs.2013.03.003

Benko, H., and Feiner, S. (2007). "Balloon Selection: A Multi-finger Technique for Accurate Low-Fatigue 3d Selection," in IEEE Symp. on 3D User Interfaces (IEEE). doi:10.1109/3DUI.2007.340778

Bowman, D. A., and Hodges, L. F. (1997a). "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments," in Proceedings of the 1997 Symposium on Interactive 3D Graphics (New York, NY, USA: ACM), 35. doi:10.1145/253284.253301

Bowman, D. A., Johnson, D. B., and Hodges, L. F. (1999). "Testbed Evaluation of Virtual Environment Interaction Techniques," in Proc. ACM Symp. on Virtual Reality Software and Technology (VRST) (New York, NY, USA: Association for Computing Machinery), 26–33.

Bowman, D., Kruijff, E., LaViola, J., and Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Boston): Addison-Wesley.

Cashion, J., Wingrave, C., and LaViola, J. J., Jr. (2012). Dense and Dynamic 3d Selection for Game-Based Virtual Environments. *IEEE Trans. Vis. Comput. Graphics* 18, 634–642. doi:10.1109/tvcg.2012.40

Cribbie, R. A., Fiksenbaum, L., Keselman, H. J., and Wilcox, R. R. (2012). Effect of Non-normality on Test Statistics for One-Way Independent Groups Designs. *Br. J. Math. Stat. Psychol.* 65, 56–73. doi:10.1111/j.2044-8317.2011.02014.x

de Araujo e Silva, F. B. (2015). "Increasing Selection Accuracy and Speed through Progressive Refinement," Ph.D. thesis (Virginia Polytechnic Institute and State University).

de Haan, G., Griffith, E. J., Koutek, M., and Post, F. H. (2006). "Hybrid Interfaces in Ves: Intent and Interaction," in 12th Eurographics Symp. on Virtual Environments, 109–118.

de Haan, G., Koutek, M., and Post, F. (2005). "Intenselect: Using Dynamic Object Rating for Assisting 3d Object Selection," in *IPT/EGVE*, 201–209. doi:10.2312/EGVE/IPTEGVE2005/201-209

Elmqvist, N., and Tudoreanu, M. E. (2006). "Evaluating the Effectiveness of Occlusion Reduction Techniques for 3d Virtual Environments," in Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 9–18.

Elmqvist, N., and Tudoreanu, M. E. (2007). Occlusion Management in Immersive and Desktop 3d Virtual Environments: Theory and Evaluation. *Int. J. Virtual Reality* 6, 21–32.

Grossman, T., and Balakrishnan, R. (2004). Pointing at Trivariate Targets in 3d Environments. in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA: ACM, 447–454. doi:10.1145/985692.985749

Grossman, T., and Balakrishnan, R. (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, New York, NY, USA: ACM, 3–12. doi:10.1145/1166253.1166257

Hajri, A. A., Fels, S., Miller, G., and Ilich, M. (2011). "Moving Target Selection in 2d Graphical User Interfaces," in *Human-Computer Interaction – INTERACT 2011*. Editors P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler (Berlin, Heidelberg: Springer), 141–161. doi:10.1007/978-3-642-23771-3_12

Hoffmann, E. R. (1990). Capture of Moving Targets: a Modification of Fitts' Law. *Ergonomics* 34, 211–220. doi:10.1080/00140139108967307

Hornbæk, K., and Hertzum, M. (2007). Untangling the Usability of Fisheye Menus. *ACM Trans. Computer-humam Interaction* 14. doi:10.1145/1275511.1275512

Hurtienne, J., and Naumann, A. (2010). "Benchmarks for Intuitive Interaction with Mobile Devices," in Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (New York, NY, USA: ACM Press), MobileHCI'10 401–402. doi:10.1145/1851600.1851685

Jagacinski, R. J., Repperger, D. W., Ward, S. L., and Moran, M. S. (1980). A Test of Fitts' Law with Moving Targets. *Hum. Factors* 22, 225–233. doi:10.1177/001872088002200211

Kabbash, P., and Buxton, W. A. S. (1995). "The "Prince" Technique: Fitts' Law and Selection Using Area Cursors," in Proc. of the SIGCHI conference on Human factors in computing systems, 273–279.

Kopper, R., Bacim, F., and Bowman, D. A. (2011). "Rapid and Accurate 3D Selection by Progressive Refinement," in *3D User Interfaces*, 67–74. doi:10.1109/3DUI.2011.5759219

Kopper, R., Bowman, D. A., Silva, M. G., and McMahan, R. P. (2010). A Human Motor Behavior Model for Distal Pointing Tasks. *Int. J. human-computer Stud.* 68, 603–615. doi:10.1016/j.ijhcs.2010.05.001

Liao, H., Li, Y., Brooks, G., and Gordon (2016). Outlier Impact and Accommodation Methods: Multiple Comparisons of Type I Error Rates. *J. Mod. App. Stat. Meth.* 15, 452–471. doi:10.22237/jmasm/1462076520

Lu, Y., Yu, C., and Shi, Y. (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) (New York, NY, USA: IEEE), 35–43. doi:10.1109/VR46266.2020.00021

MacKenzie, I. S. (2013). A Note on the Validity of the Shannon Formulation for Fitts' Index of Difficulty. *OJAppS* 03, 360–368. doi:10.4236/ojapps.2013.36046

MacKenzie, I. S., and Buxton, W. (1992). "Extending Fitts' Law to Two-Dimensional Tasks," in Proceedings of the SIGCHI conference on Human factors in computing systems. (ACM) (New York, NY, USA: ACM Press), 219–226.

MacKenzie, I. S. (1992). Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-computer interaction* 7, 91–139. doi:10.1207/s15327051hci0701_3

MacKenzie, I. S. (2018). *The Wiley Handbook of Human Computer Interaction, Fitts' Law*. (Hoboken, NJ: Wiley), 17, 349–370. doi:10.1002/9781118976005

Mendes, D., Medeiros, D., Cordeiro, E., Sousa, M., Ferreira, A., and Jorge, J. (2017). "Precious! Out-Of-Reach Selection Using Iterative Refinement in Vr," in 2017 IEEE Symposium on 3D User Interfaces 3DUI (New York, NY, USA: IEEE), 237–238. doi:10.1109/3DUI.2017.7893359

Mine, M. R. (1995). "Virtual Environment Interaction Techniques,". Technical report (Pennsylvania, PA, USA: University of North Carolina, Department of Computer Science).

Mircioiu, C., and Atkinson, J. (2017). A Comparison of Parametric and Non-parametric Methods Applied to a Likert Scale. *Pharmacy* 5, 26. doi:10.3390/pharmacy5020026

Murata, A., and Iwase, H. (2001). Extending Fitts' Law to a Three-Dimensional Pointing Task. *Hum. Mov. Sci.* 20, 791–805. doi:10.1016/S0167-9457(01)00058-6

Naumann, A., and Hurtienne, J. (2010). "Benchmarks for Intuitive Interaction with mobile Devices," in Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (New York, NY, USA: ACM), 401–402. doi:10.1145/1851600.1851685

Olwal, A., Benko, H., and Feiner, S. (2003). "Senseshapes: Using Statistical Geometry for Object Selection in a Multimodal Augmented Reality System," in Proc. 2nd IEEE/ACM International Symp. on Mixed and Augmented Reality ISMAR '03 (New York, NY, USA: ACM Press), 300–301.

Olwal, A., and Feiner, S. (2003). *The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality*, 81–82.

Pierce, J. S., Forsberg, A. S., Conway, M. J., Hong, S., Zeleznik, R. C., and Mine, M. R. (1997). "Image Plane Interaction Techniques in 3d Immersive Environments," in Proc. Symp. on Interactive 3D Graphics I3D '97 (New York, NY, USA: ACM Press), 39–ff. doi:10.1145/253284.253303

Poupyrev, I., Weghorst, S., Billinghorst, M., and Ichikawa, T. (1997). "A Framework and Testbed for Studying Manipulation Techniques for Immersive Vr," in Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 21–28.

Riege, K., Holtkamper, T., Wesche, G., and Fröhlich, B. (2006). "The Bent Pick ray: An Extended Pointing Technique for Multi-User Interaction," in Proc. 3D User Interfaces3DUI '06 (New York, NY, USA), 63–66. doi:10.1109/VR.2006.127

Sarkar, M., and Brown, M. H. (1992). "Graphical Fisheye Views of Graphs," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (New York, NY, USA: Association for Computing Machinery), 83–91. doi:10.1145/142750.142763

Shoemaker, G., and Gutwin, C. (2007). "Supporting Multi-point Interaction in Visual Workspaces," in *Proc. SIGCHI Conference on Human Factors in Computing Systems CHI '07* (New York, NY, USA: ACM Press), 999–1008. doi:10.1145/1240624.1240777

Shoemaker, G., Tsukitani, T., Kitamura, Y., and Booth, K. S. (2012). Two-part Models Capture the Impact of Gain on Pointing Performance. *ACM Trans. Comput.-Hum. Interact.* 19 (1), 34:1–28:34. doi:10.1145/2395131.2395135

Steed, A. (2006). *Towards a General Model for Selection in Virtual Environments*. New York, NY, USA: IEEE, 103–110. doi:10.1109/VR.2006.134

Teather, R. J., and Stuerzlinger, W. (2013). "Pointing at 3d Target Projections with One-Eyed and Stereo Cursors," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (ACM) (New York, NY, USA: ACM Press), 159–168.

Ullrich, D., and Diefenbach, S. (2010). "Intui. Exploring the Facets of Intuitive Interaction," in *Mensch & Computer 2010: Interaktive Kulturen*. Editors J. Ziegler and A. Schmidt (Munich, Germany: Oldenbourg Wissenschaftsverlag), 251–260. doi:10.1524/9783486853483.251

Vanacken, L., Grossman, T., and Coninx, K. (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*, 117–124. doi:10.1109/3DUI.2007.340783

Wingrave, C. A., Bowman, D. A., and Hodges, L. F. (2005a). "Baseline Factors for Raycasting Selection," in Proceedings of HCI International.

Wingrave, C. A., Bowman, D. A., and Ramakrishnan, N. (2001). "A First Step towards Nuance-Oriented Interfaces for Virtual Environments," in Proc. Virtual Reality International Conference, 181–188.

Wingrave, C. A., Tintner, R., Walker, B. N., Bowman, D. A., and Hodges, L. F. (2005b). "Exploring Individual Differences in Raybased Selection: Strategies and Traits," in IEEE Proceedings. VR 2005. Virtual Reality, 2005, 163–170.

Wolf, D., Gugenheimer, J., Combosch, M., and Rukzio, E. (2020). "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 (New York, NY, USA: Association for Computing Machinery), 1–10. doi:10.1145/3313831.3376876

Wonner, J., Grosjean, J., Capobianco, A., and Bechmann, D. (2012). "Starfish: a Selection Technique for Dense Virtual Environments," in Proceedings of the ACM symposium on virtual reality software and technology, 101–104.

Wyss, H. P., Blach, R., and Bues, M. (2006). iSith - Intersection-Based Spatial Interaction for Two Hands. In Proc. 3D User Interfaces. 3DUI '06 (New York, NY, USA: IEEE), 59–61. doi:10.1109/VR.2006.93

Zhai, S., Buxton, W., and Milgram, P. (1994). "The "Silk Cursor": Investigating Transparency for 3D Target Acquisition," in Proc. SIGCHI conference on Human factors in computing systems CHI '94 (New York, NY, USA), 459–464. doi:10.1145/191666.191822

Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y. (2003). "Human On-Line Response to Target Expansion," in Proc. Conference on Human factors in computing systems, 177.

Zhao, H., Jin, X., Shen, J., and Lu, S. (2009). Fast and Reliable Mouse Picking Using Graphics Hardware. *Int. J. Comp. Games Tech.* 2009, 1–7. doi:10.1155/2009/730894

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.