Universität Bremen

Fachbereich 3: Mathematik und Informatik

LenSelect - Dynamic Object Scaling in Virtual Environments for Object Selection

Waldemar Wegele

Student Number: 2812630

For the attainment of the academic degree of
Master of Science in Computer Sciences

August 2020

First Supervisor: *Prof. Dr. Gabriel Zachmann*
Second Supervisor: *Prof. Dr. Johannes Schönning*
Advisor: *Dr. Rene Weller*

Contents

To this day selection in 3D-Environments is a challenge with no completely satisfactory solution. Especially overlapping objects and reduction due to perspective pose a difficult problem.
A new selection technique called “LenSelect” is discussed and evaluated in this thesis. LenSelect was examined in different test scenarios, which were designed to cover most use-cases. It’s capability was assessed against selected state-of-the-art selection techniques in terms of effectiveness and user-experience. This assessment was conducted with a head-mounted-display, the HTC Vive and a spatially-immersive-display, the Powerwall.
LenSelect is based on a consideration of Fitts’ Law, which states that the time needed to make a selection depends on the distance of the selection pointer to the object and the width of the object in direction of said pointers movement. This observation is the basis for LenSelect, as it increases the size of objects and therefore the target width to make selection faster and easier.
In a study it was found that LenSelect performs on par with IntenSelect, a state-of-the-art selection technique. And both of them perform better than RaySelection.

1. Introduction

1.1. Goals of this Thesis

LenSelect, a new kind of selection technique in virtual reality, which is based on an observation of Fitts' Law, shall be evaluated in this thesis. It will be tested with a head-mounted-display (HMD), the HTC Vive Pro and a spatially-immersive-display (SID), the Powerwall.
The proposed test scenarios are supposed to cover all use cases, however objects standing close together, so called "cluttered environments" and overlapping objects shall be examined in particular. The selection of far away objects as well as moving objects will also be examined and results directly compared to other state-of-the-art selection techniques.

1.2. Motivation

Selection is one of the most basic interactions in a virtual environment [39, p. 2]. Via the selection the user chooses one or more targets for further interaction.
Research has shown that there is no superior selection technique for all situations. Selection depends on the specific requirements of the task, the layout of the environment and experiences and preferences of the user. For example experience with video games (or rather 3D selection) and other cognitive features. Other factors like tiredness, feedback, sight of the environment, the following interaction technique or "fun" can influence the requirements of the selection technique [12, p. 635].
Therefore a variety of selection techniques have been developed throughout the years and existing techniques refined and developed further.

1.3. Definition

In 1979 Foley proposed that an interaction task for a 2D GUI can be broken down into basic interaction tasks. Following this approach Bowman proposes four universal interaction tasks for virtual environments [8, p. 37]
These proposed interaction tasks are:

1. Viewpoint Motion Control (Navigation)

2. Selection

3. Manipulation
4. System Control

Viewpoint Motion Control refers to the task of orienting and positioning ones viewpoint in a virtual environment. In VR head tracking is usually used to determine the viewpoint orientation, so the user is mostly concerned with translation, or in other words navigation through the virtual environment [8, p. 37].
Selection refers to the designation of objects by the user [8, p. 42].
Manipulation describes setting the rotation and position, and possibly other characteristics such as shape or color, of a selected object [8, p. 42].
System Control is comprised of other commands to accomplish work within the application (such as delete the selected object, save the current location, load a new model, etc.). Bowman also notes that, at a low level, System Control tasks can be characterized as selection and/or manipulation tasks [8, p. 38].
In this thesis only selection tasks will be considered.

2. State-of-the-Art

2.1. Fitts' Law

Fitts' Law describes a formula that can predict the amount of time a human needs for the selection of a target in regards to distance to the target and width of the target. "In the decades since Fitts' original publication, his relationship, or law, has proven one of the most robust, highly cited, and widely adopted models to emerge from experimental psychology." [35, p. 93]. Over the years Fitts' Law played a central role in empirical studies of the human as an information processor [35, p. 93]. The formula looks as follows:

$$MT = a + b \cdot ID \tag{2.1}$$

Where MT needed time for a selection, a the intercept in seconds and b the slope in bits per second [34, p. 353]. These are typically empirically determined. ID describes the Index of Difficulty, how difficult a given selection is.

$$ID = log_2 \left(\frac{2A}{W} \right) \tag{2.2}$$

Where A amplitude, the distance of the object to the pointer, W width of the object in movement direction of the pointer. Fitts called this amplitude the "movement tolerance", the error the user is allowed to make to still select the object [34, p. 350]. Throughout the years different forms of this equation have been developed and published. MacKenzie proposes one such other form of the formula which he calls Shannon formulation:

$$ID = log_2 \left(\frac{A}{W} + 1 \right) \tag{2.3}$$

"In 1989, it was shown that Fitts deduced his relationship citing an approximation of Shannon's theorem that only applies if the signal-to-noise ratio is large. [...] Besides the improved link with information theory [...] the Shannon formulation provides better correlations compared to the Fitts [...] formulation." [34, p. 354].
However this derivation is not without controversy as Drewes argues: "MacKenzie claims that his formula [...] shows better correlation values for experimental data. This seems to be true [...] However, correlation does not tell much and the correlation

gets even better if adding 2 instead of 1 [...]." [16, p. 8]. Drewes goes on to say: "MacKenzie criticizes Fitts' introduction of factor 2 to and argues that adding 1 instead of multiplying with 2 will guarantee positive values for the ID. He refers to Shannon's theorem 17, a formula given as a footnote in Fitts' publication, which has the desired +1 and then he does a 'direct analogy' without further explanation. In his analogy the bandwidth (measured in bits/second) shall be analog to time (measured in seconds) and power shall be analog to amplitude (power is proportional to the square of the amplitude). There is no justification for such analogy." [16, p. 8] In fact there is a debate about useful- and correctness of all of these formulations [17] [25] [37], but even though the derivation is controversial, over the years it proved itself as the most popular and widely used form of Fitts' Law [16, p. 8].
So, despite all this MacKenzies formula is used in this thesis.

2.1.1. Determining the Distance

There seems to be a bit of confusion on how amplitude and width are measured. According to Buxton and MacKenzie the minimum for A is $\frac{W}{2}$, otherwise the pointer is inside the object [36, p. 220]. This means A is the distance from the pointer's position to the edge of the object plus half the width of the object.
However they also go on to say: "If the targets are circles (or perhaps squares), then the 1D constraint in the model remains largely intact (because the "width" of a circle is the same, regardless of the angle of measurement). However, if targets are rectangles, such as words, the situation is confounded. The amplitude is still the distance to the center of the target; but the role of target width is unclear." [36, p. 220]
This can only be the case if we assume the amplitude to be measured from the pointer's position to the center of the object. However, if we assume the amplitude to be the distance of the pointer's position to the border of the object plus half the object's width in movement direction of the pointer, the parameter's stay one dimensional and are captured accurately no matter the shape of the object (s. Figure 1.
This is how the parameters are calculated in this thesis, with the exception that the distance ends at the object's border.

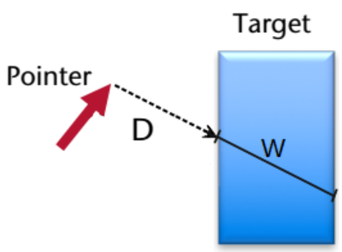


Figure 1.: Determining the distance for Fitts' Law

Then there is also the case of partially occluded movement paths for the pointer. When another object occludes the target object, the occluded parts, of course, can not be selected and therefore should not add to the width. The easiest way would be to just add up all the parts along the cursor's movement in which the object can be selected and count this as the object's width.
But there might be a psychological effect at play here. Users might subconsciously weigh these distinct selection parts and decide on which is easiest to select, taking into account distance and width of the respective parts. They might than target this specific part, even going as far as ignoring other selectable parts of the target. These other parts could be selectable earlier, before the user targeted part or they could be ignored when the user overshoots the desired part and instead opts to correct in the opposite direction.
Sadly, to the author's knowledge, there have been no studies on this possible phenomenon. And of course this is also just speculation. So in this thesis the parts are added up to generate an effective target width.

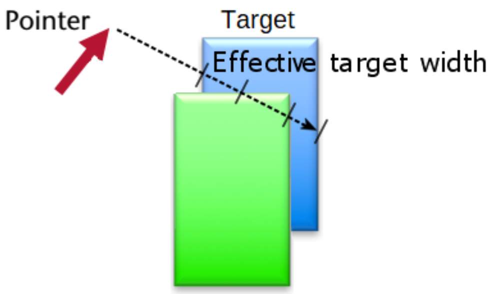


Figure 2.: Determining the Effective Target Width

2.1.2. Index of Performance

Fitts' idea encapsulates two notions, that the difficulty of a selection task can be quantified by interpreting it as information, using bits as measurement and second that the act of selecting a target is similar to transmitting information through a channel [34, p. 350].
Fitts called this transmission rate the Index of Performance, though nowadays the term throughput is used [34, p. 450].

$$IP = \frac{ID}{MT} \tag{2.4}$$

With ID measured in bits and MT measured in seconds, so the unit of the Index of Performance is $\frac{bits}{s}$. A higher IP means a better Performance, as more data is "computed" in the same amount of time. Fitts theorizes that with changes of ID, MT changes accordingly, so the IP stays constant [34, p. 351].

2.1.2.1. Index of Performance Adjusted for Accuracy

However it can be argued that this only is an approximation of what happens during a selection [34, p. 358]. Not only is there the assumption that participants instructed to select targets quickly and accurately, can balance the demands of the task correctly. But it also fails to take into account the user's attitude. If a user slows down to focus on accuracy the task changes, the same can be said for a user that focuses on acquisition speed [34, p. 357]. "In summary, Fitts' Law is a model for rapid, aimed movements, and the presence of a nominal yet consistent error rate in participants' behavior is assumed, and arguably vital [34, p. 357]."
Crossman proposes a normalization of target width by calculating the standard deviation of cursor endpoints when selection ended for the participant. "The idea was first proposed by Crossman in 1957 in an unpublished report [...]. Use of the adjustment was later examined and endorsed by Fitts [...] [34, p. 355]." Under this assumption movement amplitudes are analogous to "signal" and endpoint variability is analogous to "noise". The theorem that builds the basis for Fitts' Law assumes that the signal is "perturbed by white noise", so the analogous requirement in motor tasks is a normal distribution of hits. A property, according to MacKenzie, observed by numerous researchers [34, p. 355].

$$IP_e = \frac{ID_e}{MT} \tag{2.5}$$

$$ID_e = log_2\left(\frac{A}{W_e} + 1\right) \tag{2.6}$$

The entropy in a normal distribution is $log_2\left(\sqrt{2\pi e} \cdot \sigma\right) = log_2(4.133 \cdot \sigma)$. Splitting the constant 4.133 into a pair of z-scores, one finds that 96% of the total area is bounded by $-2.066 < z < 2.066$. So the assumption is that 96% of hits fall within the target, while 4% miss, according to MacKenzie [34, p. 355].
There are now two methods to determine effective target width. Either by calculating the standard deviation for all endpoints or via the discrete-error method [34, p. 355].
If the standard deviation is known just multiply it by 4.133.
The discrete-error method is used when only the error percentage is known. It uses a table of z-scores for areas under the unit-normal curve. The method goes as follows: If n percent of errors are observed for a given condition, determine z such that $\pm z$ contains 100 - n percent of the area under the unit-normal curve. Then multiply W by $\frac{2.066}{z}$ to get the effective target width W_e (not to be confused with the effective target width in Figure 2) [34, p. 355].
This way users that try to make an accurate selection are rewarded, since target endpoint variability is reduced, the effective target width also diminishes which leads to a rise in the Index of Difficulty [34, p. 355].
This technique dates back to 1957, but according to MacKenzie has been largely ignored in the research that followed [34, p. 356].

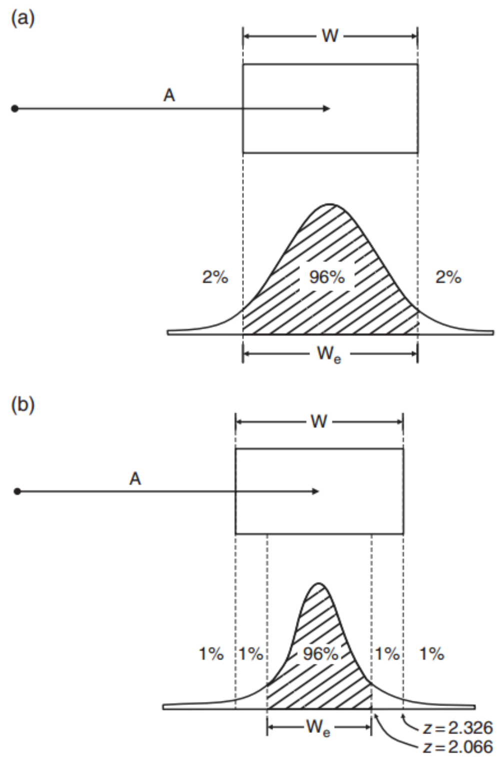


Figure 3.: Method of adjusting target width based on distribution of selections. Source: [34, p. 356].

2.1.2.2. Criticisms

Zhai argues these equations are ill-defined, however. His, very simple, argument is that, since both variables in IP depend on the value of ID ($IP = \frac{ID}{MT} = \frac{ID}{a+b \cdot ID}$), these definitions cannot be constants. A change in ID will alter the result of IP, this means a comparison can not be drawn between selection techniques.
So Fitts' Law studies should instead rely on and report both a (intercept) and b (slope), since they are true constants [62, pp. 791–792]. Zhai discusses three possible interpretations of how the throughput TP can be obtained according to ISO 9241-9.

$$TP_a = \frac{ID_{Mean}}{MT_{Mean}} \tag{2.7}$$

$$TP_b = \frac{1}{b} \tag{2.8}$$

$$TP_c = TP_{Mean} = \frac{1}{N} \sum_{i=1}^{N} \frac{ID_i}{MT_i} \tag{2.9}$$

While TP_a first calculates the means of all selection times and IDs and then calculates the throughput from there, TP_c calculates the throughput for every single

selection and obtains the mean throughput from there [62, p. 795].
But since the above observation applies to both of them they may change with the experimental setup. A direct comparison is no longer possible. [62, p. 797].
Zhai proposes three remedies for this problem:

1. Using a standard ID set in the study.
2. Using both a and b in modeling and TP_b as a matter of convenience for throughput or IP.
3. Excluding all non-informational aspects of pointing in the model.

Solution one is not feasible for this study, as the basis of LenSelect is exactly the change in ID that is supposed to lead to easier selection. Solution two is the easiest approach and seems to be favored by Zhai as well: “If we need to use the concept of throughput or index of performance, it should be defined as a simple inversion of the information coefficient b, and be used together with the a parameter, the information-independent aspect of pointing [62, p. 805].”

2.1.2.3. Consequence

As shown there are a lot of problems with and different viewpoints on what throughput even is and how it should be calculated. It is also of note that, to the author's knowledge, most of the time IP or TP is not included in the analyses of selection techniques. Most papers don't even mention them.
For the purposes of this study Zhai's solution two seems to be the most suitable. Additionally an IP in the form of TP_c is also included for comparison's sake, however it is not adjusted for pointing accuracy.

2.1.3. Two-part formulation of Fitts' Law

In 2012 Shoemaker, Tsukitani, Kitamura and Booth described limitations of one-part models of Fitts' Law when it comes to different levels of control-display gain (s. p. 14 Isomorph vs Non-Isomorph Mapping) [48]. To account for this two-part formulations of Fitts' Law have been proposed. Welford et al. suggest that two control processes need to be considered. A motor, or distance-covering phase and a visual, homing-in phase [59, p. 10].

$$MT = a + b \cdot log_2(A) - c \cdot log_2(W) \tag{2.10}$$

Shoemaker et al. reformulate this equation to match the form of equation 2.1.

$$MT = a + b \cdot log_2\left(\frac{A}{W^k}\right) \tag{2.11}$$

Where $k = \frac{c}{b}$, which allows A and W to have a different impact on the equation and model the two control processes this way. [48, p. 7].
Since the selection techniques used in this thesis have a control-display ratio of 1:1 these equations will not be used.

2.1.4. Fitts' Law in Higher Dimensions

The above form of Fitts' Law only accounts for one dimensional selection tasks, though it is often applied to 2D target selection as well. In cases like this the movement angle of the selection is ignored, instead the selection is reduced to a one dimensional selection by only considering the pointer's distance to the target and the width of the target in movement direction of the pointer. Since both of these are measured along the same axis, the formula stays one dimensional [36]. This of course is dependent on the user's perspective in a virtual 3D environment.

2.1.4.1. 2D-Selection

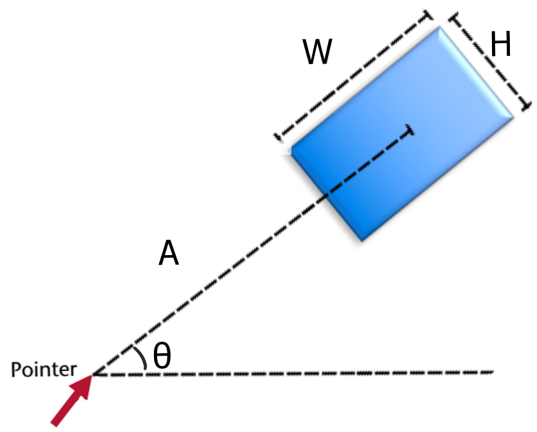


Figure 4.: 2D-Selection

It was found by Murata and Iwase that the selection of targets in the direction above the pointer took longer than that of lower and horizontal targets [41, p. 802]. This can possibly be attributed to the effect of gravity. Iwase and Murata conclude that therefore the current Model of Fitts' Law is not suited for real-world 3D pointing tasks [41, p. 802].
Iwase and Murata propose an empirically derived model, that exclusively models the effects of θ.

$$ID = log_2\left(\frac{A}{W} + 1\right) + c \cdot sin(\theta) \tag{2.12}$$

Where c is an arbitrary constant to be determined through linear regression [41, p. 800].

Expanding on the findings of Zhai and Accot [1] Grossman and Balakrishnan propose an euclidean model that can model the effects of θ separately per dimension [20, p. 450].

$$ID_{WtEuc\theta} = log_2\left(\sqrt{f_W(\theta)\left(\frac{A}{W}\right)^2 + f_H(\theta)\left(\frac{A}{H}\right)^2} + 1\right) \tag{2.13}$$

2.1.4.2. 3D-Selection

Formula 2.13 can be easily extended for 3D-pointing tasks.

$$ID_{WtEuc\theta} = log_2\left(\sqrt{f_W(\theta)\left(\frac{A}{W}\right)^2 + f_H(\theta)\left(\frac{A}{H}\right)^2 + f_D(\theta)\left(\frac{A}{D}\right)^2} + 1\right) \tag{2.14}$$

2.1.4.3. Problems:

All these models use empirically determined constants that are easy to calculate "after the fact". Formulas 2.13 and 2.14 are also complex for a simple task, while not taking into account objects that are not shaped like rectangles/boxes or rectangles/boxes whose dimensional axes might not be aligned to the coordinate axes.
While Formula 2.12 is independent of the object's alignment it's not clear if the angles impact should be applied to the Index of Difficulty or as a separate factor exclusively to determine selection time. It is also not clear how much the weighting constant might differ from person to person, as muscle mass coupled with the weight of the pointing device would be important considerations if gravity is the deciding factor.
In the end using Fitts' formula for one dimensional tasks seems like the better approach, since most objects in this thesis will be underneath the pointer anyway.
The formulae seem either too complicated while not taking into account all possibilities or the problem they try to solve does not come up regularly enough to warrant their inclusion in this thesis.

2.1.5. Fitts' Law for Moving Targets

Even though research into Fitts' Law for moving targets is already rather old (Jagacinski, Repperger, Ward and Moran researched it in 1980 [29]), resources are scarce.
Multiple papers derived different equations, often building onto the formulas found by Jagacinski, et al. and evaluated their merit sometimes with the data of Jagacinski, et al. and sometimes with user data from different experiments.
It is clear that the Index of Difficulty must be modified to capture the time needed for the selection of moving objects [29, pp. 229–230]. In Jagacinski, et al.'s publication [29] the following formula is proposed among others:

$$MT = a + b \cdot A + c \cdot (V + 1) \cdot \left(\frac{1}{W} - 1\right) \tag{2.15}$$

Where V is the object's velocity and c another empirically derived constant.
This formula apparently is a good fit for the data presented, however as the authors mention, the data comes from a small sample size and the given equation is just an approximation for this data [29, p. 231]. In their experiment an object was considered selected if the cursor stayed inside it for 350 ms. The authors also propose another equation to consider this capture time duration T_c.

$$MT = a + b \cdot \log_2\left(\frac{2A}{W}\right) + c \cdot log_2\left(\frac{V}{\frac{W}{T_c}} + 1\right) \tag{2.16}$$

This model is very different to equation 2.15 and simply reduces to Fitts' Law for $V = 0$. But also provided a worse fit for their data. A third model was proposed as well, but also failed to provide a better fit [29, p. 232].
Using a first order control system Hoffmann derived an ID for moving targets closer in form to Fitts' original ID.

$$ID = log_2\left(\frac{A + \frac{V}{K}}{\frac{W}{2} - \frac{V}{K}}\right) \tag{2.17}$$

Where V is dependent on the movement direction of the target, negative when the target is approaching the pointer and positive else, Hoffmann defines an approaching target as having "negative" velocity in his paper [24, p. 211]. He also provided a two-part formulation for his model.

$$MT = a + b \cdot \log_2\left(A + \frac{V}{K}\right) + c \cdot log_2\left(\frac{W}{2} - \frac{V}{K}\right) \tag{2.18}$$

In both cases Hoffmann's formulas performed worse than those of Jagacinski, et al. with fits of 0.92 vs 0.97 and 0.98 vs. 0.99, although both of them still present an excellent fit. [24, pp. 217–218].
K serves to determine the critical speed V_{crit} at which deliberate target acquisition is no longer possible.

$$V_{crit} = \frac{W \cdot K}{2} \tag{2.19}$$

Hoffmann calculated the value of K in two different ways, first by regression analysis and second by observing the critical speed at the "threshold" for loss of the target object (50% of not capturing the target) [24, p. 219].

The latter being dependent on the percentage of successful captures, means outside factors can have high influence on the value.
The regression on the other hand is very specialized and again can be only calculated "after the fact".
Both are also highly dependent on the design of the experiment.
In 2011 Hajiri, Fels, Miller and Ilich again investigated different models for capture of moving targets. By modeling target acquisition time they derive a similar equation as Hoffmann [23, p. 148]. This model again providing a good fit for the given data [23, p. 156].
There of course have also been forays into 2D target acquisition for moving targets, however since the formula for stationary targets in this thesis is already one dimensional, these will not be considered. Instead Hoffmann's or Hajiri, et al's formula is used.

2.2. Selection Technique Categorization

Selection Techniques can be categorized in different ways depending on their properties.

2.2.1. Taxonomies

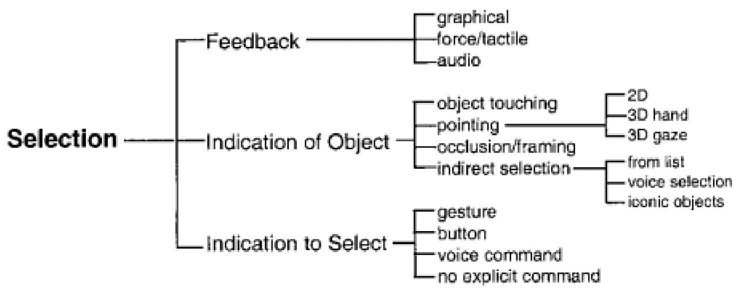


Figure 5.: Taxonomy for selection techniques
Source: Based on [10, p. 77].

2.2.1.1. Feedback

Feedback gives the user an indication if an action was successful. There is both positive (an action was successful) and negative feedback (an action was unsuccessful). There are different types of feedback as well, the most common being graphical feedback, a change in color, the object disappearing/changing etc. The second most common type is audio feedback, here a sound is played to inform the user. Force/-tactile feedback is used less often, since the hardware does not always support it. Both of these belong to the group of haptic feedback, the difference is that tactile feedback is a force felt on the fingers, while force feedback can be felt anywhere on the body. This can for example be a vibration of the controller.

2.2.1.2. Indication of Object

How is the object indicated by the user?
This can happen via simply touching the object. Or by pointing at the object. The object can be pointed at either in 2D or 3D space and also via the user's hand or by the user looking at the object.
Another possibility is to draw a frame around the object again either in 2D or in 3D via a bounding volume/area. Or the object can simply be occluded by a user controlled object.
Indirect selection describes a form of selection in which the user does not directly interact with the object, like choosing it from a list.

2.2.1.3. Indication to Select

The indication to select describes how the actual selection is triggered. Different mechanisms can have different effects on the selection outcome. For example pressing a button can lead to a tremble in the user's hand, if the object is indicated by pointing this might lead to the user missing the object and therefore lead to selection errors [61, p. 1].

2.2.2. Immediate Selection vs Iterative Refinement

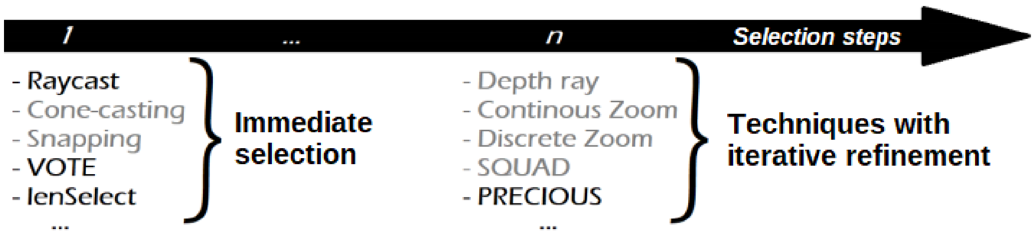


Figure 6.: Categories of selection techniques
Source: Based on [45, p. 3].

Immediate selection is self-explanatory, the highlighted object(s) will be selected as soon as the user confirms the selection. For a raycast this means the intersected object will be selected. Often these selection techniques come with a few disadvantages like reduced pointing precision (s. p. 15, Hand Jitter).
In contrast techniques with iterative refinement add extra steps to the selection. Grossman and Balakrishnan's Depth Ray, for example, adds a depth marker that can be moved along the ray by the user, thus adding an extra step to selection [21]. While Cashion, Wingrave and LaViola's SQUAD puts objects of interest determined

in a first selection step into quadrants. By iterative selection of these quadrants a single object for selection is determined (s. p. 21, SQUAD) [12, p. 636]. Since these techniques involve more steps they are usually slower than techniques with immediate selection.

2.2.3. Isomorph vs Non-Isomorph Mapping

The control-display ratio describes how translation and rotation of the input device is transferred to the selection tool [3, p. 126]. A direct mapping (1:1) between pointing device and selection tool is called isomorph, while a non-direct mapping is called non-isomorph [3, p. 126].
It's also possible to take advantage of this insight for better user interaction, as the PRISM (s. p. 27, PRISM) interaction technique shows [19].

2.2.4. Multiple vs. Single Selection

Usually most selection techniques can only select a single object at a time. Sometimes selection techniques can select multiple objects at once, these techniques typically use a volume for selection.
In this thesis only selection techniques for single selection are considered.

2.2.5. Selection Tool

There is also the question of how the selection is visualized to the user. This could be a virtual hand at the same position as the user's controller in the virtual environment. A ray, a cone or more elaborate means. Some of these could even be invisible to the user, a selection technique could use a raycast to select an object, while not actually showing the cast ray to the user.

2.2.6. Disambiguation

Some selection techniques also use a disambiguation mechanism to determine the object the user most likely wants to select. IntenSelect (s. p. 20, IntenSelect), for example, calculates a score for all objects according to their angle to the pointing device and hovers the object with the highest score for the user to select. [22, p. 204].

2.3. Problems of 3D Selection

There are some additional problems further complicating the selection in 3D environments.

2.3.1. Hand Jitter

As 3D selection usually happens in mid-air there is also the problem of the user not being able to rest their hands on a surface, which introduces a constant shaking to the user's hand and therefore the cast ray [31, p. 604]. Different enhancements are often included to combat this. These enhancements reach from input filtering, using area cursors instead of point cursors, the ability to zoom in on an area to the ability to control cursor speed [31, p. 604]. These enhancements also add complexity which might lead the user to develop different strategies to achieve their goal. So performance might depend on whether the user chose the correct approach. [31, p. 604].
As already pointed out above, the act of pressing a button to indicate selection can also lead to hand jitter. According to Wolf, Gugenheimer, Combosh and Rukzio this effect accounts for 30.45% of errors for selection, they call it the "Heisenberg Effect of Spatial Interaction" [61, p. 1]. Their experiment also shows that smaller targets lead to a higher error in regard to this effect [61, p. 9]. Since all selection techniques used in this study are ray-based and use the same button to trigger selection they should have (roughly) the same amount of error, therefore this effect is not accounted for in this study.

2.3.2. Noise

Similar to hand jitter the tracking of a given device can also introduce a "shakiness" to the virtual tool. Even when laying the device down on a table tracking noise may be observed in the virtual tool, depending on how precise the tracking is. It's impact is, of course, also highly dependent on the severity of said noise.

2.3.3. Eye-Hand Visibility Mismatch

Another problem is the mismatch between the objects that are (un)occluded for the user's eyes and the user's hand, for techniques where the ray originates at the user's hand. Objects might differ in "visibility" for these differing positions. [4, p. 43]. These issues have mostly been ignored in contemporary research and it's effects on selection performance have not been studied in depth [4, p. 43]. Another problem of this is that the user might point at the object directly and not it's visible screen projection, which means a further decrease in accuracy [4, p. 44].

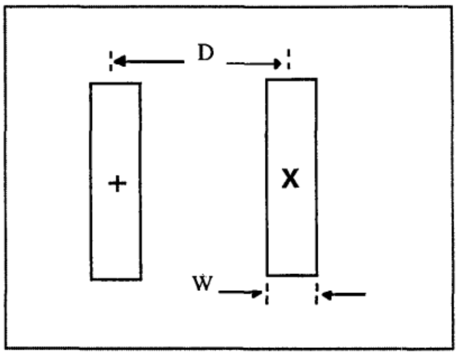
2.4. Standards and Guidelines

Even though research in this part of human-computer-interaction has a long history, experiments usually don't follow many formal guidelines. There are general guidelines for how experiments should be set up, but they often are ignored in formal studies. Standard test scenarios, on the other hand, are hard to come by. Usually researchers opt for primitives like cubes and spheres as objects for the selection.

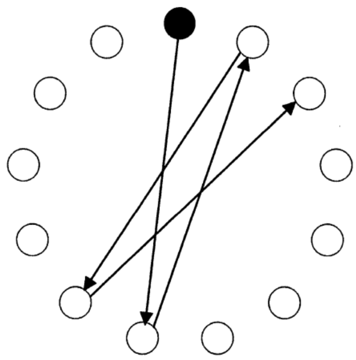
Sometimes more complex objects are used to simulate an actual use case or a more natural environment, like fruits in a fruit stand. Sometimes the objects move by themselves or stand on another object that moves/rotates.
How the experiment is conducted is often not clear as well. Like how, or if, the distance between initial pointer position and the object was calculated. Did the pointer need to be at a certain location before the selection could be started? Did the user get a moment to get accustomed to the scene? How long was that moment, how was it measured/accounted for in the experiment? These questions often go unanswered.
Bowman, Johnson and Hodges propose test beds to evaluate selection techniques for universal interaction tasks [10, p. 78]. They propose a methodology based on taxonomies, performance metrics and outside factors [10, pp. 76–78].
Poupyrev, Weghorst, Billinghurst and Ichikawa identify a list of different factors that should be considered in the measurement of a selection technique's performance. They call these factors task parameters [44, p. 22]. These parameters can depend on different sources, like the user, the device, the interaction technique, the application and even the task context [44, pp. 22, 23].
Both Bowman et. al. and Poupyrev et. al. propose additional performance criteria. Not only completion time and error rate should be considered in the experiment but also additional factors typically found in the research field of human-computer-interaction. Such as: Ease of use, ease of learning and presence, as these will also be important to the success of an interface [10, p. 78], [44, p. 24].
A lot of different factors should be considered, to make selection techniques truly comparable, however most experiments don't seem to take these conclusions into consideration.

2.4.1. ISO 9241

ISO 9241-9 evaluates user performance, comfort and effort in an attempt to define a procedure for selection technique evaluation. The standard recommends using the Index of Performance as the sole measure of performance for a selection technique. It uses the IP_e formulation for it's Index of Performance
Furthermore the standard provides six tasks: one directional (horizontal) tapping, multi-directional tapping, dragging, free-hand tracing, free-hand input (hand written characters or pictures) and grasp and park (homing, device switching). The tasks used should be determined by the intended use of the device [15, p. 216]. It also provides a questionnaire [15, p. 217].
Douglas, Kirkpatrick and MacKenzie endorse the standard. According to them in recent years the effective width has replaced the measured width in the computation of the Index of Performance. However the authors feel that separate measures of speed as movement time and accuracy as error rate can not be replaced by a single measure. They also criticize the standard for not incorporating learning effect. The standard recommends a sample size of 25, they however claim 12 for each between-subject condition is standard practice for pointing device performance experiments.



One-directional selection task
Source: [15, p. 216]



Reciprocal selection task
Source: [51, p. 50]

Figure 7.: Selection of selection tasks proposed by ISO 9241-9

In addition, apparently the 7-Point Likert scale was finer than participants could distinguish [15, pp. 220–221]. It is of note that the standard was written for one and two dimensional pointing tasks.
The standard was superseded by ISO 9241-411 in 2014, but still uses the same definition for throughput and the same selection tasks.
As already mentioned in Chapter 2.1.2.2 Criticisms on p. 7 the throughput proposed by this standard is problematic. In addition the selection tasks are not well suited for 3D selection, since they weren't designed for it. The questionnaire is written to assess usability but I feel a more general questionnaire specific for measuring usability, like QUESI, is more suitable for this task.

2.4.2. User Personalization

LaCoche, et al. investigated the use of machine learning to select the selection technique best suited to the user's needs and preferences. Their results show that the user's preference can have a big impact on performance [32, p. 48]. It is not hard to imagine that further personalization of parameters of a specific selection technique could also have an impact. Of course adjusting all these factors by hand for a study is excessive and not feasible, additionally it makes comparison between selection techniques impossible.

3. Related Work

3.1. Selection Techniques Used in the Study

In this section the different selection techniques used in the final study of this thesis will be introduced. Of course this is only a small selection of all the selection techniques that have been developed through the years. The selection techniques described here are all pointing based, an iterative selection technique is also included.

3.1.1. Raycast

Casting a ray is one of the most simple and widely used selection techniques [12, p. 643]. A ray is cast from the input device into the scene, the hit object is selected when the user activates the selection. This method is quick and easy to understand for users but is problematic for the selection of small or occluded objects [12, p. 643]. Raycasting does not provide high-precision pointing with visually small targets, because small movements of the user's hand result in larger movements of the end of the ray as the distance along the ray increases [5, p. 785].
Raycast serves as a baseline in this thesis.

3.1.2. LenSelect

LenSelect is a new type of selection technique for virtual environments. It is based on the idea of dynamically scaling up targets depending on their proximity to the projected point from the users pointing device [45, p. 13]. This point is projected via a raycast. In a certain radius around the hit point of this raycast a percentagewise enlargement is made, this way the object's width gets bigger, therefore, according to Fitts' Law the selection is easier and faster [45, p. 13].
Cluttered environments and the selection of far away objects are LenSelects designated use-cases. LenSelect is intuitive since its basic premise does not differ much from a simple raycast and the basic selection is still the same.
However the implementation from the bachelor thesis comes with a few flaws. Floating objects for example, do not profit from this technique at all, since without any reference objects for the ray to hit the object will be resized only after it was already hit by the ray. Objects with a difficult selection context, like objects behind a hill, where it's hard to get the hit point close enough to the object are also problematic. Different vectors of approach of the ray also lead to different results. If a big object is in front of the target and the ray approaches from the front, the target could

be occluded by said object. If the ray approaches from behind the target object however, that same object will not be scaled and therefore won't occlude the target. Also in cluttered environments scaling the objects up can lead to higher occlusion between objects than before the scaling.
So in this thesis the "lens shape" (e.g. the bounding volume in which objects are scaled) was reworked and different scaling algorithms were tested in a prestudy.

3.1.3. IntenSelect

IntenSelect was designed to tackle three main aspects: Selection accuracy, selection ambiguity and selection complexity [22, p. 2].
IntenSelect uses a cone where the tip rests inside the pointing device and it's base points in the direction of the ray emitted by the pointing device. For all objects in the scene an "inside"-test is performed [22, p. 204]. For this test the vector between the middle point of the object and the tip of the cone is calculated. With this the angle between the ray and this new vector is compared to the opening angle of the cone. If the angle is smaller than the cone's opening angle the object is inside the volume [22, p. 204].

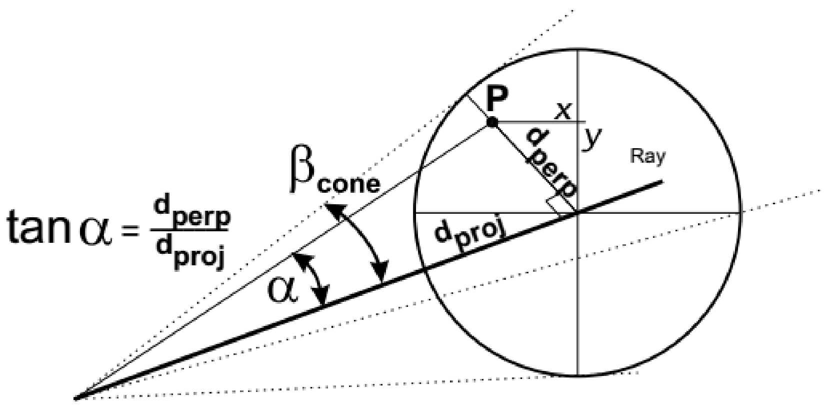


Figure 8.: Determining whether point P is inside the selection volume.
Source: [22, p. 204].

For all objects inside the cone a score based on the relation between the angle between object and ray and the cone's opening angle is calculated. The object with the highest score is then selected.
One of the simplest scoring functions would be:

$$s = 1 - \frac{\alpha}{\beta} \tag{3.1}$$

Where, α the angle between the ray and the vector to the middle point of the object and β the opening angle of the cone.

However "during testing trials it quickly became obvious that, using this scoring metric, it is easier to select distant objects than it is to select nearby objects [22, p. 204]. That is because the use of α implies a growing distance perpendicular to the ray the further away the object is along the vector between the tip of the cone and the middle point of the object.
So the authors introduced a scoring function that took this discrepancy into account:

$$s = 1 - \frac{atan\left(\frac{d_{perp}}{{d_{proj}}^{k}}\right)}{\beta} \tag{3.2}$$

With d_{perp}, the distance of point to the ray and d_{proj}, the distance of the object to the tip of the cone along the ray and k, a compensation constants [22, p. 205].
With similar scores a fast switching behavior between objects can be observed [22, p. 205]. So additionally a score accumulation over time was implemented.

$$s_{contrib}(t) = 1 - \frac{atan\left(\frac{d_{perp}(t)}{d_{proj}(t)^{k}}\right)}{\beta} \tag{3.3}$$

$$s_{total}(t) = s_{total}(t-1) \cdot c_s + s_{contrib}(t) \cdot c_g \tag{3.4}$$

The contribution score remains unaltered only it now denotes the score at time step t. The total score is defined by the total score of the previous time step and the current contribution score. Both are multiplied by constant scaling factors c_s and c_g, these typically define the "stickiness" and "snappiness" of the selection.
The total score is progressively reduced by the c_s factor so scores of previous frames get slowly faded out [22, p. 205].

3.1.4. SQUAD

"SQUAD is a selection technique that uses progressive refinement for narrowing the choice of objects to select from." [12, p. 636]. This is done by doing a sphere trace and arranging copies of all objects contained in it on the screen in quadrants. With each progressive selection the objects get rearranged into quadrants until only one is left, this object will finally be selected [12, p. 636].
The strong suit of this selection technique is that even highly occluded objects can be easily selected, a disadvantage is that selection becomes difficult if there are multiple objects that look the same and therefore can not be differentiated by their looks alone [12, p. 636]. Depending on the amount of objects in the sphere trace, selection times can also be long [12, p. 636].
SQUAD is not used directly in this study, rather it's evolution Expand is.

3.1.5. Expand

Expand is a refinement of SQUAD and an iterative approach to object selection. It tries to alleviate the problem of losing the original context of the objects. This was mostly done so it is easier to discern similar looking objects [12, p. 636]
Instead of quadrants copies of the objects inside the sphere cast are aligned on a grid on the screen. This grid is dynamically arranged depending on the number of objects that need to be placed in it and fills the screen [12, p. 636].
Although one of it's goals is to keep original context between objects, the copies don't seem to be sorted in a way to keep their relationship intact. A possible sorting could, for example, be their location in the x-y plane. So the implementation described in the paper is inefficient, in that the user will have to search for their target in the grid. It also seems like the user still has to point at object's in the grid, this is inefficient, as pointing at the cell containing the object would lead to better selection times. However the original implementation is used in this thesis.

3.2. Further Selection Techniques

A bigger selection of researched selection techniques, that are not used in this thesis, is discussed here.

3.2.1. Ballon Selection

The metaphor used is a helium balloon on a string, this string is held by the user's hands. The dominant hand moves the position of the balloon (the other hand is following along). While the second hand changes the balloon's height by moving the string closer or farther away from the dominant hand. The object overlapping the virtual sphere will be selected [7, p. 80]. This way a 3DOF positioning task can be broken down into a 2DOF position task on the ground plane and a 1DOF task to control the balloons height [7, pp. 80–81].
Ballon Selection showed less errors than the Wand technique and only slightly more than a simple keyboard. At the same time the technique is about as fast as the Wand technique, as the experiment shows [7, p. 84].

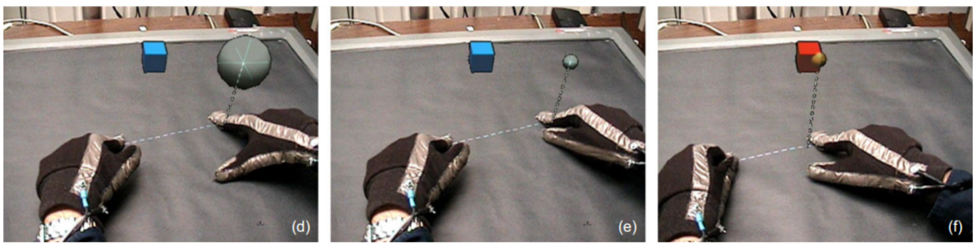


Figure 9.: The Balloon Selection technique.
Source: Based on: [7, p. 81]

3.2.2. Bubble Cursor

Bubble Cursor uses a volume cursor in the form of a sphere, the object inside the volume will be selected. At the same time this activation volume is required to be unambiguous, so the sphere dynamically resizes such that only one object falls inside it. If a single object can't be captured within the bubble then an additional bubble is rendered around the closest object. A cross hair is drawn in the center of the sphere to indicate it's current position [57, p. 119]. Similar to depth ray a marker can be moved up and down the ray emitted by the pointing device. Distances to objects are calculated relative to this marker. With this the user can also select occluded objects [57, p. 120].

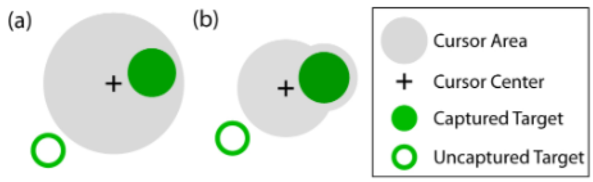


Figure 10.: Bubble change depending on surrounding targets.
Source: Based on: [52, p. 281]

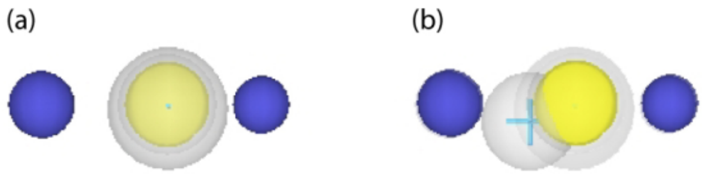


Figure 11.: Bubble change depending on surrounding targets. Extended to 3D.
Source: [57, p. 119]

The circle's size is determined by calculating an intersecting and containing distance for all targets $T_1, T_2, ...T_n$.
Intersecting Distance i ($IntD_i$):
The length of the shortest distance between the center of the circle and any point on target T_i.
Containing Distance i ($ContD_i$):
The length of the longest distance between the center of the circle and any point on the target T_i
Set i = index of closest target by intersecting distance
Set j = index of second closest target by intersecting distance
Set radius of bubble cursor circle = $min(ContD_i, IntD_j)$
If the radius is smaller than $ContD_i$ an additional circle is drawn at the center of object T_i that contains it fully [52, p. 283].

In a study Bubble Cursor performed better than Point Cursor, but slightly worse than Depth Ray [57, pp. 122–124].

3.2.3. Bubble Ray

Bubble Ray is an alternative version of Bubble Cursor for selection in 3D environments. While Bubble Cursor uses the euclidean distance between objects to calculate the distance Bubble Ray uses the angular distance instead. Lu, Yu and Shi argue depth perception in VR is too poor to use a direct representation of distance [33, p. 36]. They also altered the visual design of the bubble. It's now a disc projected on a sphere centered on the user's eyes and the disc is tangent to the user's eyes. Furthermore a curved ray to the object that will be selected was added for better target indication [33, p. 37].
The study showed slightly better performance for the angular version of Bubble Cursor over the euclidean version. It also performed better than other tested techniques and overall was preferred by users [33, pp. 39–41].

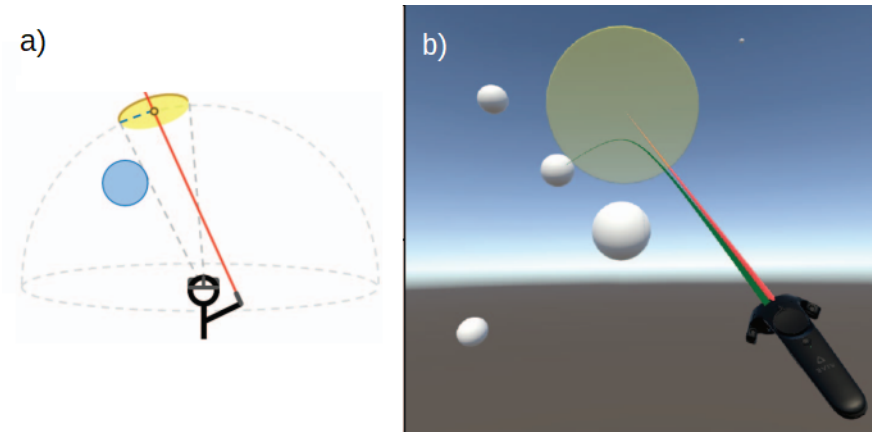


Figure 12.: a) Conceptual design of the bubble. b) Bubble Ray in the 3D environment.
Source: Based on: [33, pp. 35, 37]

3.2.4. Depth Ray

Depth Ray is basically a normal ray cast, only that a depth marker is positioned along the ray that can be moved by the user. The object that is intersected by the ray, that is closest to the depth marker, is ready for selection by the user. The depth marker is controlled by moving the hand backwards/forwards [58, p. 120].
In Grossman and Balakrshinan's experiment Depth Ray performed the best of all tested selection techniques [53, p. 10]. The other selection techniques in that study tried to improve upon the design of the Depth Ray but failed to do so.

3.2.5. Double Bubble

Double Bubble is similar to Expand, only it uses Bubble Cursor for the initial selection, instead of a sphere trace, but the sphere of the Bubble Cursor can only be reduced to a predefined threshold. If there is more than one object in the bubble Expand is used for further refinement [2, p. 61].
In Bacim's experiments Double Bubble produces, on average, less errors than Bubble Cursor and is faster as well [2, pp. 80, 81].

3.2.6. Flower Ray

Flower Ray is a two-step selection technique designed for volumetric displays. When the user pushes a button all objects intersected by a ray change their positions and "flower" out so they become unobstructed by each other, while other objects are made transparent, so as not to occlude the relevant objects [53, p. 7]. When this new "menu" appears a 2D cursor is drawn in the center of the menu. Additionally a circle is drawn around the cursor. This circle represents the minim movement distance for a selection, if the cursor is moved outside the circle the closest object to the cursor is highlighted. By letting go of the button the user selects the highlighted object [53, p. 8]. All objects intersected by the ray are colored green, while the object that will be selected is colored red [53, p. 7].
Interestingly the Flower Ray had the smallest selection time but the longest disambiguation time in Grossman and Balakrishnan's results [53, p. 10].

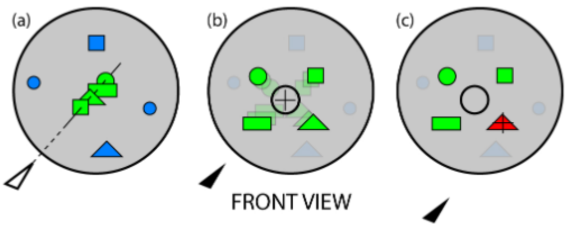


Figure 13.: The Flower Ray selection technique.
Source: [53, p. 7]

3.2.7. Hook

Hook was developed to aid in the selection of moving targets. It assists the user with a heuristic which includes time as a factor. The calculated score changes with the distance to targets. The target with the highest score is considered as "hooked" a visual feedback shows the user that this target is selectable [43, p. 120].
Each time step the the score is calculated for each object, the objects are ordered in a list of increasing distances. Scores are increased/decreased according to that list, to avoid system inertia only a number of closest targets (NCT) will have their score increased, the rest has theirs decreased. To maintain stability on hooking a small

number of NCT is beneficial. The NCT was established by user observation. It was observed that the cursor is following the target at a certain distance, the faster the target moved the bigger the distance. If this distance is considered as the possible space of deliberate target acquisition, all objects within a radius around the cursor will be considered possible targets the user wants to select [43, p. 200]. NCT "[...] is then dependent on: (1) the total number of targets (TT), (2) the spherical volume of the surrounding (SV) and (3) the global frame volume (FV) in which targets move. So: $NCT = TT \cdot \frac{SV}{FV}$. [43, p. 200]"
The added or subtracted amount of score is dependent on time and rank in the ordered list. For NCT targets equation 3.5 is used, the rest use 3.6. NCT is the number of closest targets and i the target's number in the list.

$$T_iScore(t) = T_iScore(t-1) + (NCT - i) \cdot \Delta t \tag{3.5}$$

$$T_iScore(t) = T_iScore(t-1) - \frac{NCT}{2} \cdot \Delta t \tag{3.6}$$

3.2.8. Lock Ray

Lock Ray is another technique separated into two phases. It works mostly like Depth Ray, only the user first determines the location and rotation of the ray, then while holding the button the depth marker can be moved back and forth. At this time moving the ray is not possible, it is locked. The depth marker is always placed in the center of the ray initially. The object closest to the depth marker will be selected when the button is released. Should the user miss the target when the ray is locked they can move their hand perpendicular to the ray to cancel the selection. All objects on the ray are colored green, while the current selection target is colored red [53, p. 7].

3.2.9. PRECIOUS

PRECIOUS uses iterative refinement for out-of-reach selection. It uses a cone as a selection metaphor. By rotating the wrist the user changes the opening angle and by extending or pulling back their hand user's can change the cones depth (height) [38, p. 237].
When multiple objects intersect the cone during selection the user is moved closer to them via teleportation. How the point for teleportation is determined is not described in the paper, however [38, p. 237]. If only two objects remain, a disambiguation canvas is used, where both objects are placed side by side in front of the user for easier selection. After the selection the user is teleported back to their initial position [38, p. 238].

The technique shows high ease-of-use and satisfaction according to the experiment in the paper, however users also reported high physical and visual discomfort [38, p. 238].

3.2.10. PRISM

Although designed for object manipulation PRISM is a good example for non-isomorph mapping. The idea is to use two distinct modes, one in which the control-display-ratio (CD-ratio) is increased to allow for more precise pointing, while the other provides direct and unconstrained interaction [19, p. 2].
The hand speed of the user is used to dynamically switch between these modes. As discussed in the Two-part formulation of Fitts' Law (p. 8) a selection can be divided into two control processes. The distance-covering phase tries to cover the distance to the object as fast as possible, while precise selection happens in the homing-in phase. PRISM tries to take advantage of this.

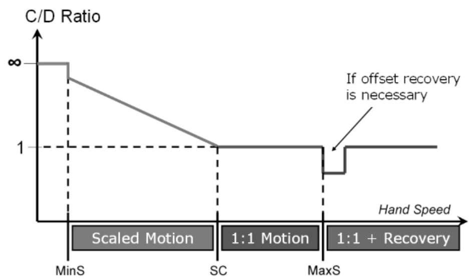


Figure 14.: CD-ratio adjustment for PRISM at different speeds.
Source: [19, p. 5]

PRISM uses three thresholds to determine the CD-ratio. A minimum speed $MinS$, this is used to filter out hand jitter, as the user is not moving their hand with purpose, motion below this speed could also be tracking error. The next threshold is the Scaling Constant SC, if the user's motion is slower than this speed the user likely has a precise goal in mind. The CD-ratio is inversely proportional to the speed at which the user's hand moves. So the closer the user is to $MinS$ the slower a manipulated object would move. The closer to SC the closer the CD-ratio resembles a 1:1 mapping. SC is the speed at which the CD-ratio becomes 1. A third constant $MaxS$ also triggers offset recovery, used to let the object catch up to the hand, of course this is not needed if PRISM is used as a selection technique.

3.2.11. Ray Cursor

Ray Cursor acts similar to Bubble Cursor, it uses a depth marker that the user can move along the ray.
Even though described as a new selection technique it is basically Bubble Cursor with filtering of hand jitter and a non-linear transfer function for the movement of

the depth marker.
Additionally different ways of highlighting the object to select were investigated as were different gain functions for the marker movement. For the filtering the 1€-Filter was used [6, pp. 3–4].
The cursor speed is calculated as follows:

$$v_{cur} = g(v_{pad}, d_{cur}) \cdot v_{pad} \tag{3.7}$$

Where v_{cur} the current speed of the marker, v_{pad} the speed of the contact point on the pad in m/s, d_{cur} distance between hand and cursor in m and g the used gain function.
Two gain functions were examined one based on finger speed, which uses a higher gain at high speeds and a lower gain at low speeds:

$$VitLerp(v_{pad}) = \begin{cases} k1, & \text{if } x \leq v_1 \\ k2, & \text{if } x \geq v_1 \\ k1 + \frac{k2-k1}{v2_v1} \cdot (v_{pad} - v_1), & \text{otherwise} \end{cases} \tag{3.8}$$

The other depends on cursor position:

$$DistDep(d_{cur}) = k \cdot \sqrt{d_{cur}^2 + d^2} \tag{3.9}$$

Both were investigated in a separate experiment as was a combination of the two [6, p. 4].
Highlighted objects show less selection errors, while RopeCursor + Highlight had the lowest selection time in the experiment shown in the paper [6, p. 6].
VitLerp showed the lowest error rate of all gain functions while having the lowest selection times, or at least selection times on par with the combination of both gain functions [6, p. 7].

3.2.12. Shadow Cone

Shadow Cone is designed for the selection of multiple objects. An implicit bounding volume is created by the user by moving a cone attached to their hand. All objects that are always inside the cone for the duration of the motion, and therefore this implicit bounding volume, are selected. The user presses a button, then targets the objects, if they release the button the objects are selected [49, p. 166].
This is implemented as follows: When the user presses a button the target set is initialized with all objects within a target angle of the ray from the hand. Each frame any object who's angle is no longer inside the target angle is removed from the set. Objects are not re-added to this initial set. When the user releases the button all objects within the target set are selected [49, p. 166].
The technique performed about on par with the normal Cone Selection for head-tracked tasks and worse for the non-head-tracked tasks [49, p. 168].

3.2.13. Smart Ray

Smart Ray memorizes the intersection of a single ray over a length of time. The disambiguation phase does not lie in the hand of the user anymore instead a predictive algorithm is used. The algorithm weights the targets based on their proximity to the target cursor. The closer the ray gets to the center of the target, the higher the weight increase will be. The intersected targets are highlighted green, while the current selection target is highlighted red [53, p. 8]. This technique takes away control from the user, but could potentially speed up selection. At least in theory, as Grossman and Balakrishnan's results show the selection, curiously, takes much longer than the other selection techniques [53, p. 10].

3.2.14. Starfish

Starfish was designed to operate in sparse and dense scenes. The name comes from the closed volume that is freely positioned by the user. Which is built on the union of several branches, starting from a pointer to a set of close targets [28, p. 102]. The user controls the head of the Starfish, the pointer from which the branches start. These branches are dynamically rebuilt while the head is moved. When one of the desired targets is reached the user can lock the shape. The head is still controlled by the user but constrained to the volume. Then the user moves the head along a branch to select the object at it's end when the button is released [28, p. 102].
For Starfish to work a set of preselected targets must be calculated. Three filters are used to achieve this:

1. **Distance Filter:** All targets whose euclidean distance is bigger than a parameter R_{max} are rejected.
2. **Angle Filter:** A small angle between two branches may lead to difficulties in entering the correct branch. So if the angle between two preliminary targets is smaller than a given Θ_{min}, the farthest is rejected. Tests show that an angle larger or equal to $\frac{\pi}{8}$ feels comfortable.
3. **Quantity Filter:** Finally only a set amount of N_{max} targets is accepted. The closest targets will be preferred.

The surfaces for Starfish's extremities are dynamically calculated at run time, with a target at the end of each [28, p. 102].
According to the paper Starfish has been deemed fun and comfortable as well as showing promising selection times. However no data of these experiments is shown in the paper [28, p. 104].

3.2.15. Go-Go

Go-Go is an arm extension technique. An area is defined at some distance from the user, inside this area the virtual hand moves at the same rate the user's hand does. When the hand is moved outside this area the virtual hand moves increasingly faster, following a non-linear, increasing function [9, p. 36]. This way the user can grab objects outside of their reach, while keeping one of the most natural selection metaphors, as the only action required are arm movement and grabbing, like in the real world [9, p. 36].

3.2.16. Wand

Wand is an incredibly simple technique. The user holds a wand in their hand, at the tip a sphere is rendered in the virtual environment. The user can change the radius of the sphere. The object inside the sphere is selected if a button is pressed [7, p. 83].

3.2.17. Virtual Hand

The virtual hand is exactly that: A virtual hand whose movements correspond to those of the user. This selection technique can be very intricate, with complete tracking of the hand including finger tips, or very rudimentary, where only the hand's position is tracked and the object is selected by a simple gesture.

3.2.18. Zoom

Zoom encompasses multiple different selection techniques, that magnify a certain portion of the screen for easier selection. Some use a fish-eye lens with certain mappings [46] to magnify the area while others may distort the object itself [11].
Zoom-and-Pick, built for hand-held projectors, was designed to alleviate jitter and resolution limitations. It uses a square fish eye lens, with adjustable zoom level, centered on the pointer. The widget also follows the pointer. To address the jitter problem a dead zone within the bounds of the lens is defined, as long as the pointer remains inside the dead zone the widget will not move [18, p. 75].
In the experiment Zoom-and-Pick shows a mostly reduced selection time compared to a regular ray cast and drastically reduced error rate [18, p. 78].

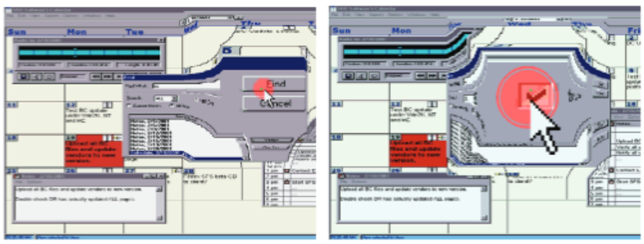


Figure 15.: Zoom-and-Pick magnification
Source: [18, p. 75]

4. Solution

A user study will be conducted to compare the different selection techniques and analyze their usefulness. A within-subjects design is used in the study, the participants will encounter nine test scenarios in which they select different objects. The test scenarios were designed with analyzing different aspects of the selection techniques in mind. Quantitative data, like selection time and Index of Difficulty, will be recorded for each selection. Additionally subjective data in the form of a questionnaire is also collected.

4.1. Study Design

In the study participants will conduct simple selections of various objects. They will fill in a questionnaire before, during and after the study. After they are finished with one selection technique they fill in the corresponding page(s) of the questionnaire.
Participants are instructed to stay in a marked square on the floor as to assure a similar distance for each participant. They are also instructed to use their dominant hand for pointing.
To avoid learning effects the latin square method [60] is used to iterate through combinations of selection techniques and test scenarios.
The participant's height can mean a change in perspective and hence overlap for the target object, however since these changes are accounted for by Fitts' Law user height is not recorded.
The experiment was designed in a way that color blind people would not have problems to participate. The highlight color for the target object (the one the participant is supposed to select) in particular was chosen to be yellow, in accordance to the perceived color spectrum shown on page 62 of [30]. As the author knows no people with color blindness however the intended effect of these measures could not be confirmed.

4.2. Study Procedure

Participants will enter a "practice mode" for each new selection technique introduced. Here they can practice how to operate the new selection technique, they can practice as much as they want, with all test scenarios available. However they can not switch selection techniques. Data is not recorded in this mode. After the participant feels comfortable to go ahead the actual study begins with the practiced

selection technique.
The target object participants are supposed to select is highlighted with a yellow border. This object will be selected first. The “ID-Helper”, a red sphere, will then appear. Participants will then select the ID-Helper, this will make the timer for the selection start. The timer stops when the target object is selected again. Selection errors will only be counted in the phase between the selection of the ID-Helper and the target object. This is to make sure the participants know where the target object is. So that only the selection time is measured and search time not included. The ID-Helper is supposed to ensure a minimum distance for the selection and a common starting point for all participants.

4.3. Test Scenarios

Unfortunately there does not seem to be an agreed upon set of standard test scenarios for selection techniques in 3D selection (other than the ones proposed by the ISO standards and those are designed for 2D pointing), therefore nine test scenarios will be introduced, each designed for testing a certain condition.
The shown test scenarios are supposed to cover all use-cases. They were designed to test different aspects of object selection. The parameters identified in [44] were used as a basis for the design of the test scenarios. These include: Distance to target object, size of the target object, occlusion, density and movement [44, p. 23]
Figure 16 on page 34 shows pictures of all used test scenarios.

Propane Tanks Close and Far
Two of the scenarios use five propane tanks sorted in two rows, three tanks in the front and two in the back alternating between both rows. One of the scenarios is placed farther away from the user than the other. These are designed to test the selection techniques for low density environments. By comparing both of these with each other it is also possible to get an idea of how suitable the technique is for distant-pointing tasks.

Floating Spheres
Twelve rotating spheres were placed a certain distance from a center point so that they form a circle. These spheres rotate around said center point. The idea is to have slow moving objects that are also predictable in their movement. This scenario is similar to the standard test scenario shown in ISO 9241 in Figure 7 on page 17.

Miscellaneous
This scenario uses all kinds of different forms and sizes of objects, to replicate a more natural context for a selection. The objects have different sizes and dimensions and are more or less randomly placed but also in a way to simulate a more or less cluttered environment.

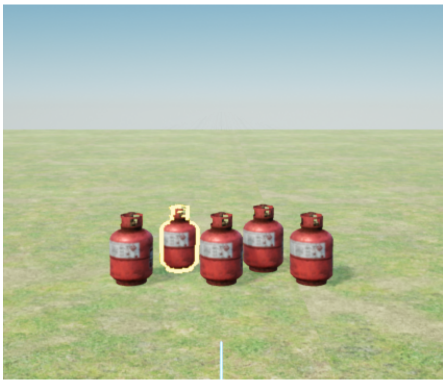
Erratically Moving Spheres
In a certain area ten spheres move around slowly but also randomly change direction, these spheres can overlap a lot. The idea here is to have moving objects, that are also unpredictable in their movement.

Fast Moving Single Sphere
A single sphere moves fast in a certain area. Like the erratically moving spheres this sphere is unpredictable in it's movement, but since it is a single sphere it will not be overlapped by other objects.

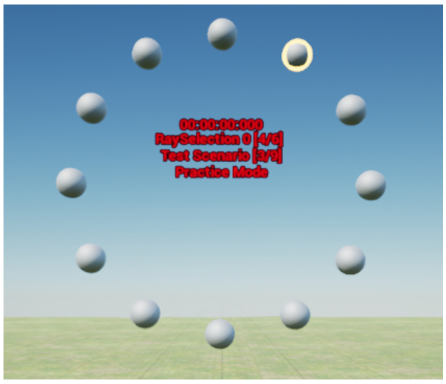
Stacked Cubes
An amount of cubes were put in a line with two more rows stacked upon each other. The goal here is to simulate medium occlusion.

Densely Placed Cans
Three rows of cans were put closely behind each other, these should simulate high density selection with no movement.

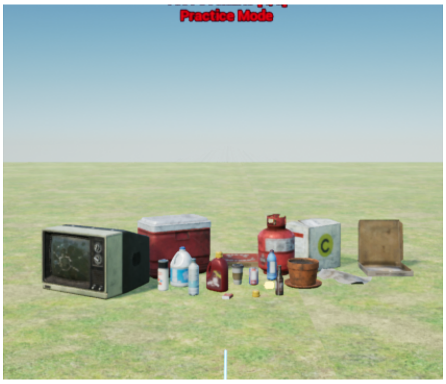
Rotating Cans
Multiple circles of cans were placed atop a rotating table. The goal is to simulate a high density selection with slow movement.



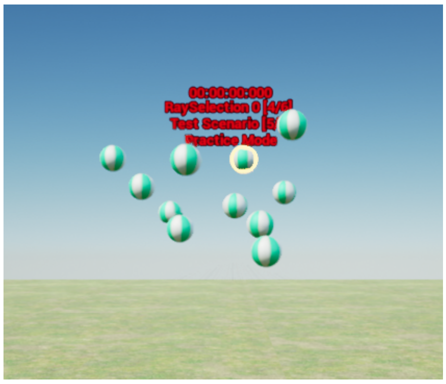
Propane tanks for close and far selection.



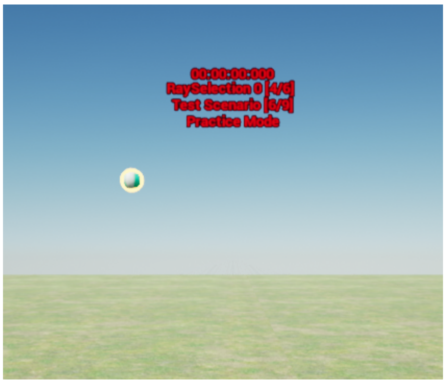
Rotating spheres.



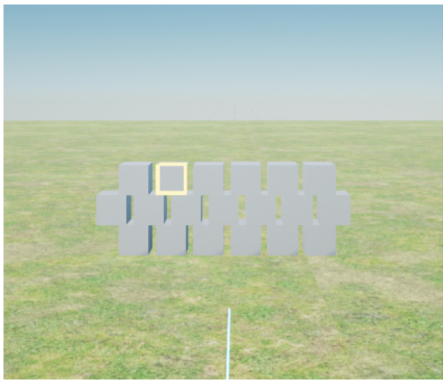
Miscellaneous objects.



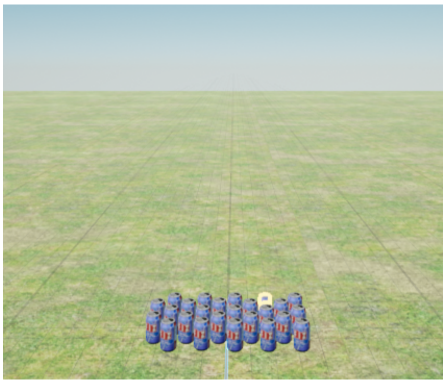
Spheres moving in an erratic fashion.



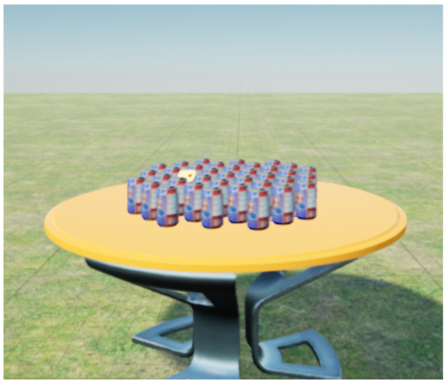
Single, fast moving sphere.



Stacked cubes.



Rows of cluttered cans.



Cans on a rotating table.

Figure 16.: The different test scenarios used in the study.

4.4. Performance Analysis

The study consists of two parts, the data collected during selection of the objects and a questionnaire for the users to fill out.

4.4.1. Fitts' Law Analysis

For this analysis three values are recorded: Time needed for the selection, the Index of Difficulty and the errors per selection, in accordance with [44, p. 24]. This data is collected per selection technique and test scenario, so it's possible to compare how selection techniques fared with different test scenarios. Additionally the Index of Performance is calculated from the selection time and the Index of Difficulty.

4.4.2. Questionnaire

As suggested in [44, p. 24] and [10, p. 78] data on how the user feels about the selection technique is also gathered. These include: Ease-of-use, ease-of-learning and sense of presence. QUESI is used for this purpose in this study.
QUESI is designed to measure the subjective consequences of intuitive use and user satisfaction with a product [42, p. 401]. It connects intuitive use with effective interaction by the user [42, p. 401], hence it poses a good basis for the user's thoughts on 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/Preknowledge and Perceived Error Rate [42, p. 401]. The mean of corresponding questions is the score for each scale. A single score, the QUESI Total Value, can be used for direct comparisons. It's equal to the mean of all five scales [42, p. 401].
The questions: "How complex did you find the selection technique?" and "Using the selection technique is fun." were added as additional items to the questionnaire. QUESI uses performance-based and cognitive measures to quantify intuitive use [54, p. 60]. As stated in [54, p. 46] an intuitive system surpasses user expectations, interaction with it feels special, magical in a way. This describes the sense of presence mentioned in [44, p. 24] pretty well and is included in the INTUI questionnaire. QUESI lacks a measure like this. So this additional scale was added to the questionnaire.
According to INTUI four items correspond to this scale [54, p. 78], however one of these items directly asks if the interaction felt magical to the participant. This question is vague, as "Magical Experience" is a fuzzy term and therefore would need to be explained to participants beforehand. Additionally definition and interpretation of the term can vary wildly, this makes comparison difficult. For this reason the item is removed from the questionnaire and only the remaining ones are used.
In addition to all this the user is asked to rank the different selection techniques directly. Where first lace is the favorite selection technique and last is the least

favorite selection technique. For calculation in the evaluation each technique will get a certain amount of points according to it's rank. The least preferred selection technique gets one point, the second to last gets two and so on.
Furthermore age, dominant hand, color blindness and profession of the user is documented. In a prequestionnaire the participant is asked if they have experience with the task at hand, with VR in general and if they can use their dominant hand without problem.
A postquestionnaire is used to determine if a participant had problems with the study itself or the system used to conduct the study.
The questionnaire used can be found in Appendix 4.4.2. Questionnaire on page 35.

5. Implementation

Unreal Engine version 4.21.2 was used to realize the implementation of the experiment. Scene and objects were adopted from Florian Rohde's bachelor thesis, although the test scenario was changed.
The implementation was mostly done in C++ although some parts were also realized in Unreal Blueprints.
For this thesis multiple versions of LenSelect were developed and some dropped early on, after determining they were not promising enough. Additionally different possibilities for extra parameters were explored.

5.1. LenSelect Lens Types

The lens describes a volume, objects inside this volume will be scaled, objects outside this volume will keep their original size. The lens needs to calculate the normalized distance d_n for the scaling functions, since this is dependent on the shape of the lens the corresponding equations will be given there as well. Since objects outside the lens are not considered for the calculation of d_n it can only assume values $[0, 1]$.

5.1.1. Lens Sphere

The original LenSelect as described in Florian Rohde's bachelor thesis uses a sphere for the lens [45, p. 13]. The normalized distance for objects is calculated as follows:

$$d_n = \frac{d_s}{r_s} \tag{5.1}$$

Where d_s the distance of the closest point of the object to the center of the sphere, r_s the radius of the sphere.
In practice the spherical lens showed a number of problems as outlined in chapter 2 State-of-the-Art. This lead to switching to a conical lens.

5.1.2. Lens Cone

A cone was the natural next step for a lens volume. A multitude of other selection techniques use cones for selection, such as IntenSelect [22].

$$d_n = \frac{d_c}{r_c} \tag{5.2}$$

Where d_c the closest distance between object and ray and r_c the radius of the cone at the depth of the closest point to the object along the ray.
d_c and r_c can be determined with simple trigonometry:

$$d_c = sin(\alpha) \cdot c_1 \tag{5.3}$$
$$r_c = sin(\beta) \cdot c_2 \tag{5.4}$$

Where c_1, c_2 length of the respective Hypotenuse, α the angle between ray and the Hypotenuse and β the opening angle of the cone.
Analogous to how IntenSelect handles it's calculations (s. Figure 8 on p. 20).
All in all the conical lens feels more natural and solves most of the problems of the spherical lens (objects floating in the air benefit from the scaling, less dependent on selection context and the vector of approach of the ray makes less of a difference).

5.2. LenSelect Scaling Functions

Different scaling functions were tested in a prestudy, to determine which works best. Most of them use the normalized distance d_n for further calculations. This distance is measured in outward direction, meaning the center of the lens has a normalized distance of zero, whereas the surface of the lens has a normalized distance of one. Every object has a maximum scaling factor which can be changed independently for each object.

$$s = s_o + f_s \cdot s_m \tag{5.5}$$

Where s the final scale of the object, s_o the original scale of the object, f_s the scaling factor as determined by one of the scaling functions (between 0 and 1) for this object and s_m the maximum scaling factor for this object.

5.2.1. Linear Scaling

Linear scaling uses d_n directly:

$$f_s = 1 - d_n \tag{5.6}$$

So objects on the surface of the cone will have no scaling applied, while objects hit by the ray will have maximum scaling applied.

5.2.2. Root Scaling

The idea behind this scaling function is to discriminate stronger against objects that are farther away from the ray. In theory this would make selection easier as objects farther away from the ray will have less overlap with the object the user wants to select.

$$f_s = 1 - \sqrt[4]{d_n} \tag{5.7}$$

The fourth root was deemed to work best after some testing, as the scale falls rapidly for the first 10% of the distance but still keeps a good size for objects that fall outside those initial 10%.

5.2.3. Same-Screen-Size Scaling

The idea here is to scale all objects inside the lens so that they keep to a certain size on the screen, no matter how far away they are from the camera.

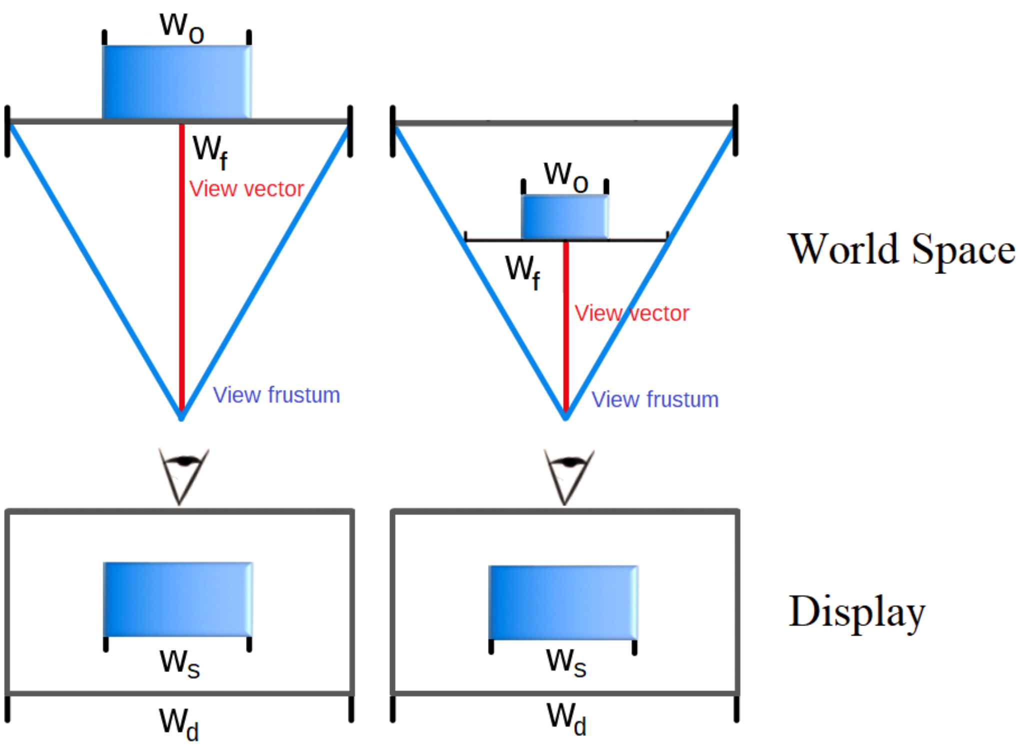


Figure 17.: Same-Screen-Size Scaling: The object is scaled so that no matter the distance to the camera it occupies the same space on screen.

For the object to keep it's size on screen the ratios between screen and object and view frustum and object need to be the same:

$$\frac{W_o}{W_f} = \frac{W_s}{W_v} \tag{5.8}$$

$$W_o = \frac{W_f \cdot W_s}{W_v} \tag{5.9}$$

$$W_f(d) = 2 \cdot tan\left(\frac{\gamma}{2}\right) \cdot d \tag{5.10}$$

Where W_o the object's width in world-space, the size to which the object needs to be scaled. W_f the width of the view frustum at depth d, W_s the object's width on the screen in pixels and W_v the width of the view port in pixels. γ the angle between the view vector and the vector between camera and object.
The width the object takes on the screen is than dependent on W_s. Since it would be tedious to adjust this value by hand for every object and for better control the additional variable "Focus Depth" is introduced. It describes the depth for which W_s is calculated for each object.

$$W_s(fd) = \frac{W_o \cdot W_v}{W_f(fd)} \tag{5.11}$$

The formula follows from equation 5.9. Basically we assume fd to be the distance at which an object keeps it's original scale. If an object is farther away than fd it will be scaled up, if it's closer it will be scaled down. Choosing *df* is also more intuitive than choosing W_s.
In another observation smaller objects benefit more from a bigger scaling than objects that are already big. This will be considered as well.

$$A_o = \frac{X_o + Y_o + Z_o}{3} \tag{5.12}$$

$$f_a = 1 + \frac{1}{A_o} \tag{5.13}$$

Where A_o the average of the extents of the object and X_o, Y_o, Z_o the objects extents. This adds an additional scaling factor in favor of smaller objects and is directly applied to $W_s(fd)$.
So in the calculation for the final scale both of these are considered:

$$W_w = W_s(fd) \cdot f_a \tag{5.14}$$

$$s = \frac{W_f \cdot W_w}{W_v} \tag{5.15}$$

Where W_w the wanted size of the object in screen-space. By inserting W_w into equation 5.9 we get the scale of the object in world-space.
Note that this scaling function does not return a scaling factor but the final scale of the object.

5.2.4. Combined Scaling

This scaling function combines the Same-Screen-Size scaling with Root scaling.

$$s = s_s - s_m + s_r \cdot s_m \tag{5.16}$$

Where s_s the object scale given by Same-Screen-Size scaling, s_m the maximum scaling factor of this object, s_r the scaling factor given by Root scaling.
Basically Same-Screen-Size scaling determines the maximum size of the object. While Root scaling determines how much of the maximum scaling factor applies to this object. The minimum scale of an object is therefore $s_s - s_m$, while the maximum scale is s_s.
Note that, again, this is the object's final scale not a scaling factor.

5.3. TelescopeSelect

TelescopeSelect was an early prototype based on the Depth Ray [21, p. 454] selection technique. With TelescopeSelect a sphere could be moved along the ray coming from the pointing device. Objects inside this sphere would be resized for easier selection. It was ultimately dropped because it was cumbersome and slow to move the sphere.

5.4. IntenSelect

Since a cone that can check for object overlaps was already implemented there was no need to check the angles for the selection volume test as suggested in [22, p. 204] All objects will be selected with the conic scoring function and temporal score accumulation. But the phase in which the user selects the ID-Helper after selecting the target object for the first time uses the normal RaySelection. This is to make sure the selection stays inside the ID-Helper, hence keeping roughly the same distance to the target for all selection techniques.
A red bending ray shows the user which object is currently hovered in addition to the usual highlighting. This ray disappears if no object would be selected. It's derived from Unreal's USplineMeshComponent and basically sets the start tangent to d_{proj} and the end tangent to d_{perp} for a given object's midpoint. So the ray originates from the pointing device and it's tip is at the midpoint of the given object.

5.5. Expand

Just as IntenSelect, Expand uses RaySelection for selecting the target and the ID-Helper before the actual selection starts.
After that a red cursor appears. A sphere-trace is performed from the camera

location through this cursor. Copies of all objects are than put on a grid, while the original objects are made transparent and inaccessible to participants. All objects that were inside the sphere-trace are selectable, while objects that were outside the sphere-trace are transparent and inaccessible as well [12, p. 636]. Participants than finally select the target object from the grid and the selection is finished.
Although this is how it's described in the paper, the purpose of putting all objects on the grid, instead of just those that were hit in the sphere-trace is unclear. As these are inaccessable anyway. The only real purpose would be to keep the original context between objects, but the object positions on the grid are random so the context is lost anyway. As keeping the original context was the primary drive in developing this selection technique [12, p. 636] these are two obvious starting points for improvements regarding this selection technique. Regardless the implementation as described in [12] is used in this thesis.

5.6. Fitts' Law Implementation

For the analysis of Fitt's Law two parameters are needed: Width of the object and distance of the pointer's initial position to the object. These parameters can be measured directly in a 2D plane and directly used in the one-dimensional form of Fitts' Law.
An easy way is to make a screen shot of the selection and count the pixels for both parameters. This is easily doable with Unreal's SceneCaptureComponent. This component is attached to the camera and dimensions and field-of-view of the camera can be applied to it. So the screen shot matches the view of the participants.

5.6.1. Modifying the Screen Shot

To better distinguish between target object, occluding objects and surroundings, Unreal's custom stencil buffer is used. Object's that are activated for this special depth buffer can be assigned a stencil value, this way objects can be distinguished and a custom color can be applied to them when the buffer is read. Occluding objects are colored cyan, target objects yellow. Everything else is colored black.

5.6.2. Calculating the Pixel Location

To count the pixels we need to know the start and endpoint for the selection.
Since the basic raycast all selection techniques use also returns the exact world coordinate of the hit point, we can use these to calculate the pixel coordinates on the screen shot for them.
To achieve this goal we project the given points on a plane in front of the camera, the distance of this plane to the camera is arbitrary as we only care for the relative distances between points. We can project the point on the plane by calculating the

vector between camera location and the given point. The projection itself is handled by a simple intersection calculation with the normal-form plane.

5.6.2.1. Calculating the Canvas Dimensions

By rotating the camera's forward vector by half of the vertical field-of-view and half of the horizontal field-of-view respectively and projecting them onto the plane as well, we know the location of the top and left points of the canvas. With these we can also calculate the dimensions of the canvas by calculating the distance between them and the midpoint (given by the camera's forward vector multiplied by the distance of the plane we are projecting onto). If we double these distances we get the height and width of the canvas, in world coordinates.

5.6.2.2. Calculating the Pixel Coordinates

Given the intersection point's relative location to the canvas's width and height, where the top left corner is (0,0) and the bottom right corner is (1,1), we can obtain it's pixel coordinates by multiplying these coordinates with the screen shot's pixel dimensions.
To achieve this we need to calculate the point's distance to the top and left points, since the plane is oriented freely in space (depending on the camera's orientation) this is not completely trivial.
So first we calculate the respective hypotenuse between the points and the intersection point. With this and the corresponding directional vector of the canvas we can determine the distance.
We can get the directional vectors directly from the camera (right vector for left point and down vector for top). By calculating the dot product between the hypotenuse and the respective vector we can get the orthogonal distance between point and vector. For example by calculating the dot product of the hypotenuse and the down vector we can determine how much of the hypotenuse points down in respect to the canvas. Keep in mind that two distinct hypotenuses are used, relative to the top and left point on the canvas.

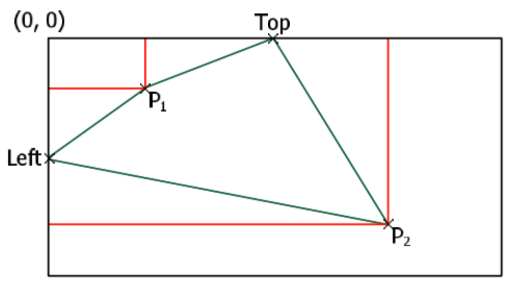


Figure 18.: Calculating the intersecting point's relative location on the canvas. Where the hypotenuses are depicted in green and the wanted distances in red for two points P_1 and P_2.

We can then obtain the percentages by dividing these distances with the dimensions of the canvas.
If we multiply these percentages with the screen shots width and height in pixels we obtain the coordinate of this pixel in the screen shot.
If any of the percentages are bigger than one or smaller than zero the point lies outside the screen shot. If the start or end point of a selection is outside the screen shot the ID can not be calculated.

5.6.3. Counting the Pixels

Counting the pixels is a simple matter of calculating the vector between start and endpoint and then checking each pixel's color along that vector in the screen shot. As long as the color is black the pixel still counts towards the distance, if the color is yellow we reached the target. Since we can't tell if we reached the end of the target or if there are more segments coming, we need to check all pixels until we reach the border of the screen shot.

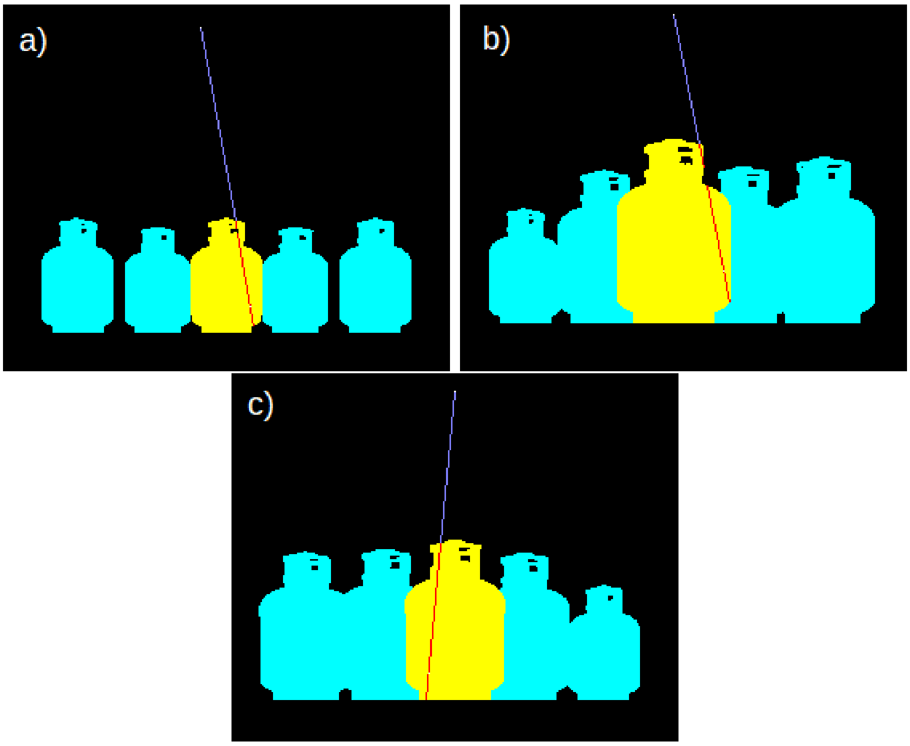


Figure 19.: Cropped debug screen shots for the ID calculation. Where the white pixels show start and endpoint of the selection, the blue line the distance and the red line(s) the target width. a) RaySelection, b) LenSelect with Root scaling, c) LenSelect with Combined scaling.

We save the beginning and end for each of these segments as a 2D vector so we can easily determine their length. Lastly we add up all of the segments inside the target

to acquire the target width.
The caveat here is, of course, that both start- and endpoint of the selection must be visible to the camera when the screen shot is taken and targets are not allowed to be partially or fully outside the screen shot. So we need additional checks for these conditions, especially since they can lead to an array-index-out-of-bounds exception for the screen shot (s. a. Figure 19 on page 44).

5.6.4. Expand

For Expand we will need the full selection path from the start point (inside the ID-Helper) to the middle point (the one where the user determines which objects will be selectable on the grid) to the endpoint (inside the target object on the grid). So we will need to save the point between start and endpoint as well.

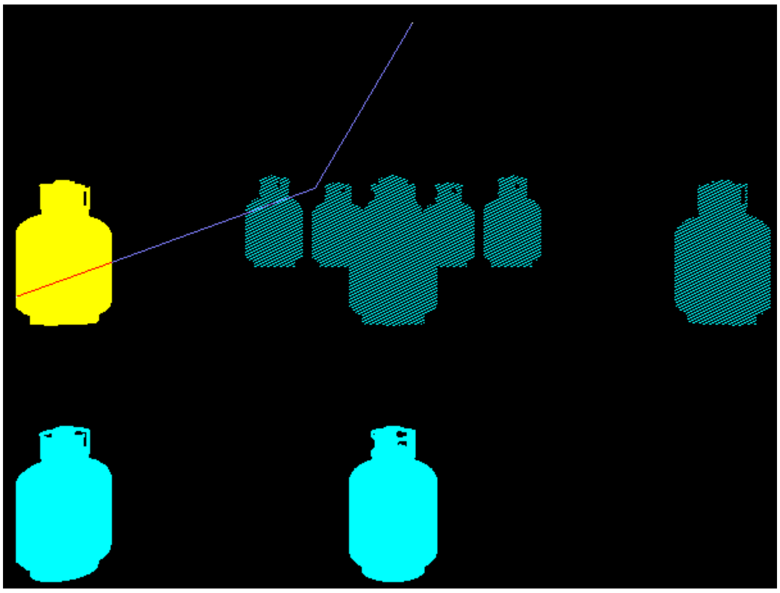


Figure 20.: Cropped screen shot for the ID calculation of Expand.

5.6.4.1. Calculating the Additional Points on the Selection Path

For this we first save the location and pointing direction of the device, at the time of each selection, for each of these additional points. When the selection is done we can calculate the intersection points with the plane the same way we did for start and endpoint and save the intersection points between them in an array to form a selection path.

5.6.4.2. Counting the Pixels for the Selection Path

This makes calculating the distance and width a bit more complicated however. We iterate through the selection points and use the current point as the start of this

segment and the following point as it's end. We will only need to check the target width for the last segment, so we can split width and distance calculation into two separate functions.
When we reach the last segment we again will have to do both the distance and width calculation. As we need to determine where the distance ends and the target begins.
The width calculation will now also have to consider that it might have already started inside the target. Apart from that the calculations stay the same.

5.6.5. IntenSelect

For IntenSelect the screen shot is insufficient, since the region in which objects become selectable is not limited to the actual objects.
Instead the screen shot is created manually and the regions for the selection calculated with IntenSelect's scoring function. This leads us to a problem, the selectable region is always relative to the pointing device and it's cast ray. There is no other solution than to create the screen shot relative to the pointing device, unlike the other moonshots which are created from the camera's view (s. a. Chapter 5.6.6 Digression on p. 48).

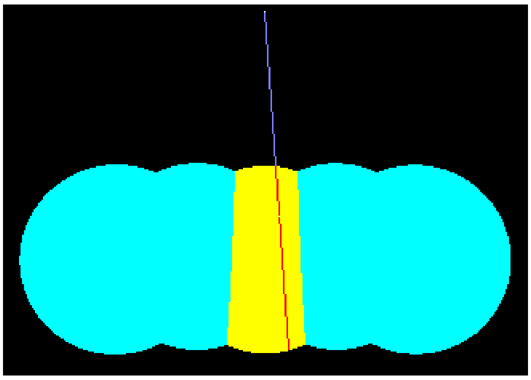


Figure 21.: Cropped screen shot for the ID calculation of IntenSelect. Note that all pixels were checked for this screen shot, not just the ones inside the target region.

5.6.5.1. Calculating the Selection's Endpoint for IntenSelect

First we need to calculate the location of the endpoint on the canvas, since the user does not have to actually point at the object we can no longer get this from the cast ray.
Instead we simulate a plane at the target object's location and orient it so that the pointing device's forward vector is orthogonal to it. By calculating the intersection point with this plane we get the endpoint for the selection.

5.6.5.2. Calculating the Target Region

We actually don't need to check every pixel, since only a small amount of pixels on the screen are of interest to us (those which may potentially lead to selection of the target object). We can do this by calculating the radius of the cone at the depth of the plane and calculating the pixels inside this circle.
Since close objects can have an effect on the target region we will also have to check against them.
To do this we project the target object's midpoint onto the canvas, next we calculate the radius of the cone for the given depth. For conic scoring the radius will be to big if we use the distance from pointing device to the plane. So instead we use the same decrease with ${d_{proj}}^k$ as in Equation 3.2 for the calculation of the angle between ray and object.
Note that the radius does not completely match the one Equation 3.2 would provide, it produces a slightly bigger radius, but having a bit of buffer in case of rounding mistakes is desirable.
With center point and radius it's easy to determine which pixels are inside the circle. First subtract the radius from the Y-component of the center point, this is the highest part of the circle. Then we only need to go through each line until we reach the line with the center point again, since we can just mirror the rest. For each line we can calculate the extents in x direction with the Pythagorean equation. We know the Hypotenuse in form of the radius and the height in form of the distance between the current line and the center point.
Lastly we just add the pixels inside the extents for this line and do the same for the mirrored line.

5.6.5.3. Calculating the Scores for each Pixel

Finally we need to calculate the scores for each pixel inside the target region, for all objects. If this pixel would select the target object color it yellow, if it would select another object color it cyan, else color it black.
To do this we will need to calculate the directional vector between the camera location (the location of the pointing device at the time of taking the screen shot) and each pixel. We calculate the top-left point of the canvas with the same method shown in Chapter 5.6.2 Calculating the Pixel Location on p. 42. We will also need to know how big a pixel on the screen shot is in world coordinates so we divide the dimensions of the screen shot in world space with the amount of pixels in both directions.
We then can construct a 3D vector representing the location of the pixel in relation to the top-left corner of the canvas. Then we orient this vector correctly by rotating it to match the pointing devices rotation in world space. With this the canvas now is orthogonal to the forward vector of the pointing device.
Lastly we calculate the point's world location by adding the canvas vector to the vector representing the top left point of the canvas in world space. We get the

directional vector by subtracting the pointing devices location from this point and then normalizing it.
With this we can use the given scoring function of IntenSelect to calculate which object would be selected by this pixel.

5.6.6. Digression

It's not fully clear if the screen shots should be taken from the pointing device's perspective or that of the camera. An argument can be made for both. Taking the screen shot from the camera's perspective makes sense, since that is the view the user's decisions stem from. At the same time the view from the pointing device shows the view for the "actual" selection. So it's the same problem as the eye-hand visibility mismatch described in chapter 2.3.3 on p. 15.
It's unclear if ID's calculated from these views are comparable. One view could potentially see things that are occluded in the other or be closer to an object. At the same time the pointing device usually is held somewhat close to the user's view point for pointing tasks.
This might, potentially, lead to a disparity between the IDs of IntenSelect and those of the rest of the selection techniques.

6. Evaluation

Two prestudies were conducted before the final study. The first to determine the most promising version of LenSelect, the second to find the most suitable parameters. All studies were conducted separately for the HMD and the Powerwall. The parameters of the test (like the distance of the objects, movement speed, etc.) were kept the same between HMD and Powerwall for each study.
The only difference for the Powerwall was a change of the ground texture, as the Powerwall has less contrast than the Vive Pro. So the grass tetxture was changed to a water texture to provide better contrast between objects and the yellow border.
Tests on the Powerwall showed that filtering for location and rotation of the input device is neccesary, because the tracking system sometimes showed latency and/or tracking issues. The 1€-Filter [13] was implemented to combat this, to keep comparability, the same filter was applied to the Vive controller with the same parameters.
After a bit of testing a “Focus Depth” of 400 was chosen for the Same-Screen-Size and Combined scaling, this value was kept throughout all studies.
A square was affixed to the ground with duct tape, participants were not allowed to step outside this square during selection. This was done so participants had an equivalent in the real world to the brown square they see in Virtual Reality. With the Powerwall participants could not see the square on the ground in virtual reality so a real counterpart was necessary. The square had a distance of 2 meters to the Powerwall.
For the sake of brevity only relevant figures are presented here, as showing all figures would lead to unreasonable bloat.

6.1. Apparatus

The HTC Vive Pro was used for the HMD study. It provides a resolution of 1440x800 pixels per eye. It was used in conjunction with Windows 10 and an NVidia Titan V with 64 GB RAM and an Intel Core i7 7800X with 3.5 GHz.
The same hardware was used for the Powerwall. Optitrack, the system used for tracking the wand, used for selection, ran on a separate computer with an Intel Core 2 Duo processor with 6700 GHz. It had 6 GB of RAM and used an NVidia GeForce GTX 680. It also ran Windows 10.
The pointing device for the Powerwall was made of the cardboard roll that comes with kitchen paper. Affixed on this cardboard roll were the markers used for the tracking. The selection was indicated via a mouse that was held in the empty non-dominant hand. This means the Heisenberg Effect of Spatial Interaction does not

apply for the Powerwall studies.

6.2. Statistical Analysis

Equivalence testing is used for further investigation into the data. The python package Pingouin [56] is used for these analyses.
The purpose is to find evidence that differences between two groups are indeed real and the data is not just differing by pure chance. To do this it demonstrates that mean differences between groups are small enough that they can be considered practically unimportant.
The null hypothesis of these tests is that populations are equal. A probability for this null hypothesis is calculated. If this probability is lower than a significance level α the null hypothesis is rejected and differences between groups are considered to be significant [50, 69f].
This α is usually chosen to be 0.05, it signifies the risk of mistakenly concluding a difference exists even though no actual difference exists between groups.
These analyses come with a few assumptions: Normality of sampling distributions of means, independent errors, homogeneity of errors and absence of outliers [50, pp. 86–90].
Both normality testing with the Shapiro-Wilk test and visualizing the data as a histogram shows that sampling distributions are not necessarily normal for Fitts' data and the questionnaire. The Levene test also shows homoscedasticity does not always apply for Fitts' data, while it mostly does for data from the questionnaire. Sample sizes can also be unequal for Fitts' data.
After some consideration Welch's ANOVA with subsequent Games-Howell post-hoc test was chosen for Fitts' data. This is because Cribbie, Fiksenbaum and Keselmann recommend the Welch test if distribution is not normal and variances are unequal [14, p. 70]. Welch's ANOVA is independent of variances but still assumes normal distribution. However: "Several studies have demonstrated that the original James and Welch procedures are generally robust (with respect to Type I errors and power) when group variances and sample sizes are extremely unequal [...] and further that the test is robust to unequal variances and non-normal data, as long as the non-normality is mild to moderate" [14, p. 57]. All of which is a concern for Fitts' data.
Welch's ANOVA is not suited for comparison of data with no variance, as that would lead to division by zero. So groups with zero variance will be removed from the test.
The Games-Howell post-hoc test is recommended after a Welch ANOVA by pingouin's documentation [55]. Like Welch's test it works with unequal variances and is robust to non-normality. But might be liberal when variances are equal and conservative when they differ, it might also be too liberal if sample sizes are too small (< 15) [47, 25f].
The Kruskal-Wallis test with subsequent Tukey post-hoc test will be used for the questionnaire. Mircioiu and Atkinson show that parametric and non-parametric

methods for Likert scale data for large sample size (> 15) perform equally well [40, p. 1].
However sample sizes are often smaller than this for the presented data, therefore the Kruskal-Wallis test is chosen. The Kruskal-Wallis test is a non-parametric test and does not assume normality, which fits Likert scale data well, but assumes equal variances. Mircioiu and Atkinson's findings, however, show for a large number of situations both normality and homogeneity of variance do not play much of a role. This applies to situations with the following characteristics: The sample sizes are (nearly) equal or the assumed underlying population distributions are of the same shape or nearly so [40, p. 11].
The Tukey post-hoc test is recommended by the pingouin library after a Kruskal-Wallis test [55]. The Games-Howell post-hoc test is not suited for this data, because it might be too liberal if sample size is too small.
Outliers as seen in the boxplots are generally not removed without good reason. Welch's ANOVA is somewhat robust to outliers [27, p. 275]. The same goes for the Kruskal-Wallis test as long as the amount of outliers is small [26, p. 468].
When a significant difference is found between groups they are connected by a red line, the amount of stars on said line shows the significance level:

* if $p <= 5e^{-2}$ and $p > 1e^{-2}$

** if $p <= 1e^{-2}$ and $p > 1e^{-3}$

*** if $p <= 1e^{-3}$

If a test could not be calculated for whatever reason a blue star is added to the figure in the upper left or beside the label.

6.3. Prestudy

The prestudy serves to determine the most promising version of LenSelect. The compared selection techniques are LenSelect with all scaling functions discussed in Chapter 5.2: LenSelect Scaling Functions, while RaySelection serves as a baseline. All nine test scenarios were used. An opening angle of 15° was used for all lenses.

6.3.1. Vive

The study had n=7 participants, all of which self-identified as male and were either students, trainees or research assistants. One person was left handed and none had a form of color blindness. Their ages ranged from 19 to 30 with a mean age of 24.86 and a standard deviation of 4.02 and a median age of 26.
Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	3.857	4	1.125
I'm often playing first person shooters.	2.714	3	1.385
I have experience with 3D-pointing devices.	3.429	3	1.294
I'm skilled at video games.	3.857	4	0.833
I have experience with HMD.	3.429	4	1.678

Table 1.: Vive PreStudy: Familiarity of participants with the presented tasks.

Participants showed a medium to high familiarity with the presented tasks and with VR in general.

Hypothesis 1: LenSelect Root will perform best of all selection techniques.
Hypothesis 2: LenSelect Root will be the preferred selection technique of participants.

6.3.1.1. Questionnaire Data

The QUESI Total is comprised of the five QUESI scales: Cognitive Load, Familiarity, Perceived Achievement of Objectives, Perceived Error Rate, Perceived Learning Effort. In addition the Magic Experience scale is also incorporated.

Complexity and Fun:

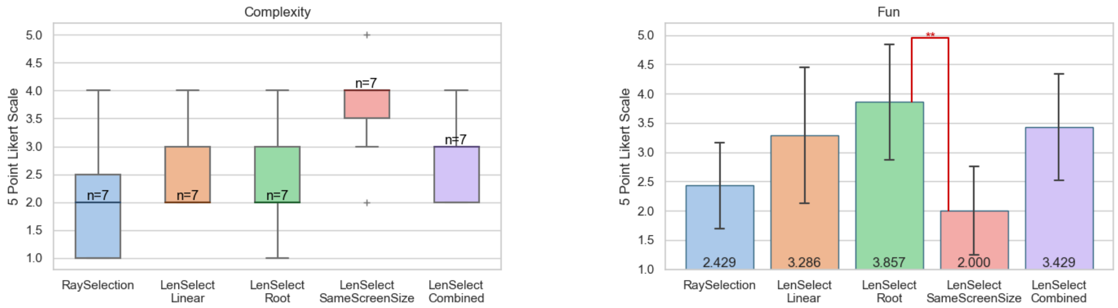


Figure 22.: Vive Prestudy: Average complexity and fun rating per selection technique. A lower value is better for complexity, while a higher value is better for the fun rating.

Root scaling is the closest in complexity to RaySelection according to participants. However, according to the Kruskal-Wallis test none of these findings are significant $H(4) = 9.21, p = 0.056$. This could be a sign that LenSelect is generally about as complex as RaySelection, no matter the scaling function. LenSelect SameScreenSize shows the highest complexity rating, while RaySelection shows the lowest. All LenSelect techniques perform similarly and somewhere between RaySeelction and LenSelect SameScreenSize.
The Kruskal-Wallis test is significant for the "fun" rating between selection techniques ($H(4) = 12.62, p = 0.013$). The Tukey post-hoc test only shows a significant difference between LenSelect Root ($M = 3.86, SD = 1.07$) and LenSelect SameScreenSize ($M = 2.0, SD = 0.82$). The former has the highest rating, the latter the lowest. Other selection techniques lie somewhere between. Still both LenSelect Linear ($M = 3.29, SD = 1.25$) and LenSelect Combined ($M = 3.43, SD = 0.98$) are rated higher than RaySelection ($M = 2.43, SD = 0.79$) even if not significantly so.

QUESI Total and User Ranking:

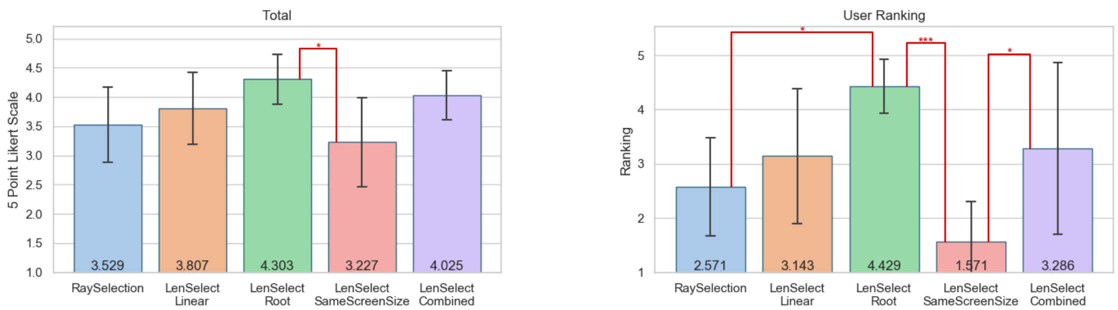


Figure 23.: Vive Prestudy: QUESI Total and user ranking per selection technique. A higher value is better.

A significant effect can be observed for the QUESI Total ($H(4) = 9.92, p = 0.042$). Post-hoc tests show that LenSelect Root ($M = 4.30, SD = 0.46$) again performs significantly better than LenSelect SameScreenSize ($M = 3.23, SD = 0.82$). LenSelect Root again performs best of all techniques. LenSelect Linear($M = 3.82, SD = 0.66$) shows a similar rating to RaySelection ($M = 3.52, SD = 0.96$). While LenSelect Combined's rating ($M = 4.03, SD = 0.46$) is closer to that of LenSelect Root.
The User Ranking shows a significant effect ($H(4) = 14.85, p = 0.005$). This time post-hoc tests reveal LenSelect Root ($4.43, SD = 0.53$) shows a significantly higher user ranking compared to RaySelection ($M = 2.57, SD = 0.98$) and LenSelect SameScreenSize ($M = 1.57, SD = 0.79$). LenSelect Combined ($M = 3.29, SD = 1.70$) shows a significantly higher rating than LenSelect SameScreenSize. LenSelect SameScreenSize performs worst of all tested selection techniques. LenSelect Linear ($M = 3.14, SD = 1.35$) shows a similar ranking to LenSelect Combined, but is not significant compared to any other technique.

Perceived Achievement of Objectives:

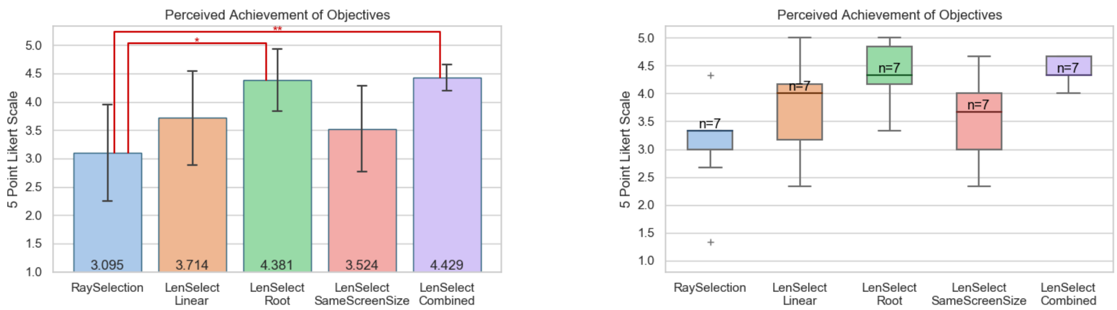


Figure 24.: Vive Prestudy: Average ranking according to Perceived Achievement of Objectives per selection technique. A higher value is better.

In the Perceived Achievement of Objectives rating ($H(4) = 13.14, p = 0.010$) both LenSelect Root ($M = 4.38, SD = 0.59$) and LenSelect Combined ($4.43, SD = 0.25$) show a significantly better rating than RaySelection ($M = 3.1, SD = 0.92$). Both are the highest rated selection techniques, with LenSelect Combined rated only slightly higher than LenSelect Root RaySelection is the lowest rated selection technique. LenSelect Linear ($M = 3.71, SD = 0.89$) and LenSelect SameScreenSize ($M = 3.52, SD = 0.81$) perform similarly and are not significant compared to any selection technique.

LenSelect Root is clearly the preferred selection technique by participants, usually followed by LenSelect Combined. LenSelect SameScreenSize on the other hand is the least preferred selection technique by a fair margin.

6.3.1.2. Fitts' Data

One outlier was removed for LenSelect Linear, due to the participant being distracted, resulting in a much higher selection time than usual.

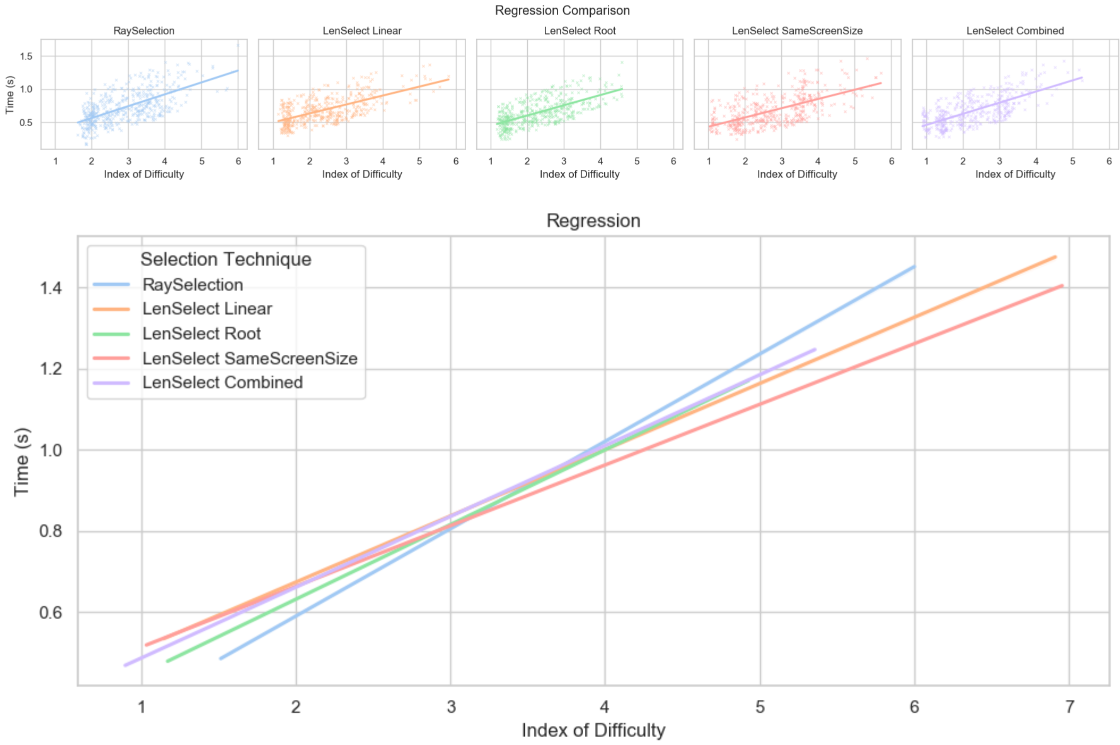


Figure 25.: Vive Prestudy: Regression Data for all selection techniques.

The scatter plots shows clusters of lower IDs for LenSelect compared to RaySelection. When plotting a linear regression for this data we can see that all LenSelect techniques start at lower IDs. LenSelect Combined and LenSelect Root also end at lower IDs, meaning these were not encountered during the study. Still LenSelect fails to produce lower selection times for lower IDs. Only at ~ 3 does LenSelect Same ScreenSize start to produce better selection times. After that LenSelect SameScreenSize provides the best selection times, followed by LenSelect Linear. For LenSelect Root and LenSelect Combined it even starts at an ID of ~ 4.
At the same these lines are not completely comparable, as LenSelect techniques would have a smaller ID compared to RaySelection for the exact same selection, due to scaling the objects.
It's of note that only samples within 95% of standard deviation of selection times around the regression line were kept for the scatter plots shown here.

Average Index of Difficulty and Task Completion Time:

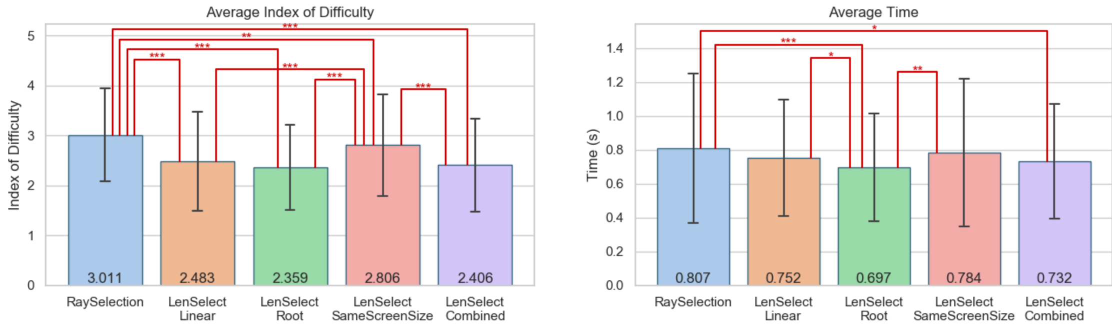


Figure 26.: Vive Prestudy: Average Index of Difficulty and task completion time per selection technique.

RaySelection ($n = 561$), LenSelect Linear ($n = 564$), LenSelect Root ($n = 572$), LenSelect SameScreenSize ($n = 563$) and LenSelect Combined ($n = 560$).
The Welch ANOVA shows a significant effect ($F(4, 1405.78) = 52.6, p < 0.001, \eta^2 = 0.07$) for the average Index of Difficulty. Where RaySelection ($M = 3.02, SD = 0.93$) produces a significantly higher ID than all other selection techniques. While LenSelect SameScreenSize ($M = 2.81, SD = 1.02$) produces a significantly higher ID than all other LenSelect techniques. The other techniques perform similarly compared to each other. With LenSelect Root ($M = 2.36, SD = 0.85$) and LenSelect Combined ($M = 2.41, SD = 0.94$) performing slightly better than LenSelect Linear ($M = 2.48, SD = 1.0$)
The average task completion time also shows a significant effect ($F(4, 1390.73) = 7.66, p < 0.001, \eta^2 = 0.01$). LenSelect Root ($M = 0.70, SD = 0.32$) performs best and significantly better than RaySelection ($M = 0.81, SD = 0.44$), LenSelect Linear ($M = 0.75, SD = 0.34$) and LenSelect SameScreenSize ($M = 0.78, SD0.44$). While LenSelect Combined ($M = 0.73, SD = 0.34$) only performs significantly better than RaySelection. LenSelect Root shows the lowest selection time. But the difference to RaySelection seems low with only $\sim 100ms$. LenSelect Combined is a close second.

Average Selection Errors and Index of Performance:
Average Selection Errors show a significant effect as well ($F(4, 1399.13) = 3.3, p = 0.011, \eta^2 = 0.006$). RaySelection ($M = 0.22, SD = 0.58$) shows significantly more selection errors than LenSelect Linear ($M = 0.13, SD = 0.46$), LenSelect Root ($M = 0.12, SD = 0.4$) and LenSelect Combined ($M = 0.12, SD = 0.38$). It's also of note here, that RaySelection shows almost double the amount of errors compared to these techniques. LenSelect SameScreenSize ($M = 0.16, SD = 0.49$) shows no significant differences.
The average Index of Performance on the other hand does not seem to make much sense in regards to the other data ($F(4, 1403.97) = 21.52, p < 0.001, \eta^2 = 0.031$). RaySelection ($M = 0.42, SD = 1.59$), which is outperformed in all other scales,

does significantly better than all other selection techniques. While LenSelect SameScreenSize ($M = 4.09, SD = 1.38$) performs better than LenSelect Linear ($M = 3.56, SD = 1.41$), LenSelect Root ($M = 3.68, SD = 1.38$) and LenSelect Combined ($M = 3.59, SD = 1.49$). Generally selection techniques that perform well in the other scales seem to perform badly here, while techniques that performed worse seem to perform better. This figure will be removed from here on out.

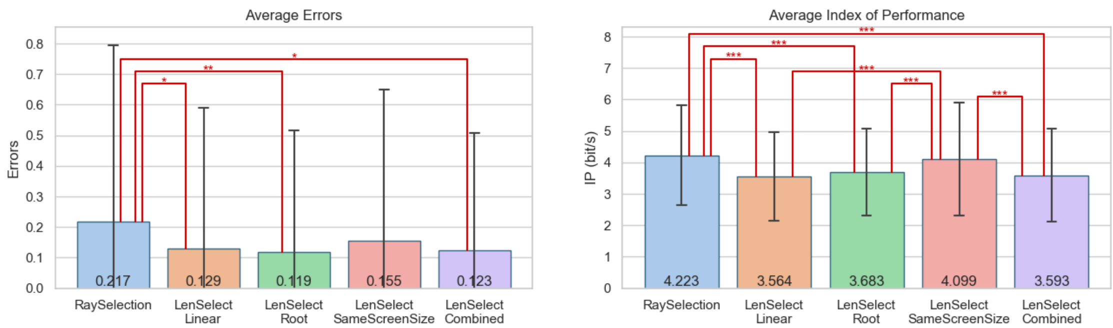


Figure 27.: Vive Prestudy: Average selection errors and Index of Performance per selection technique.

Throughput TP_b:
Even the second scale for throughput (TP_b) does not really align with the rest of the data. It's easy to tell why: TP_b only takes into account the slope of the regression and ignores that other selection techniques generally produce a lower ID. One could argue this measure is pointless then, since usually selection techniques are designed to reduce the Index of Difficulty for a selection. Both these scales will be ignored for the rest of the study.
As Zhai already argued slope and intercept are the more important factors to report. We can see that RaySelection has the lowest intercept, with the only technique coming close being LenSelect Root. The other three techniques show a similar intercept. But all LenSelect techniques show a lower slope. LenSelect SameScreenSize has the lowest slope, while LenSelect Root has the highest among LenSelect techniques.

	RaySelection	LenSelect Linear	LenSelect Root	LenSelect SameScreenSize	LenSelect Combined
a	0.210	0.366	0.297	0.297	0.294
b	0.178	0.134	0.154	0.139	0.167
TP_b	5.617	7.457	6.514	7.206	5.975

Table 2.: Vive PreStudy: a, b and TP_b for different selection techniques.

As can be seen LenSelect Combined and LenSelect Root produce, on average, a lower Index of Difficulty and less selection errors. While the difference in selection times is only minor, though statistically significant. Interestingly LenSelect SameScreenSize still performs better in these scales than RaySelection. So the dislike seen in the Questionnaire does not seem to stem form it's performance but other factors that reduce user satisfaction.

Propane Tanks Far:

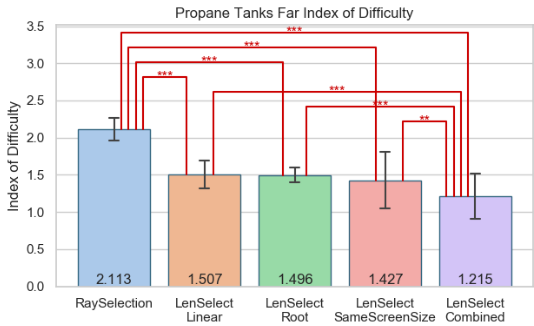
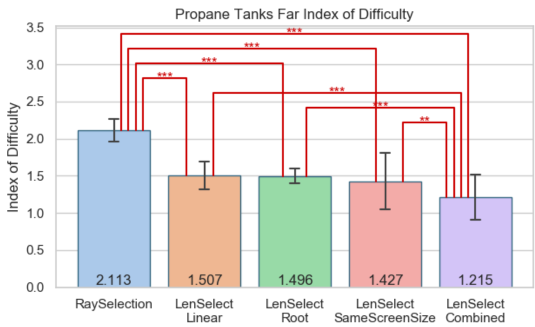


Figure 28.: Vive Prestudy: Average selection errors for the "Propane Tanks Far" test scenario.



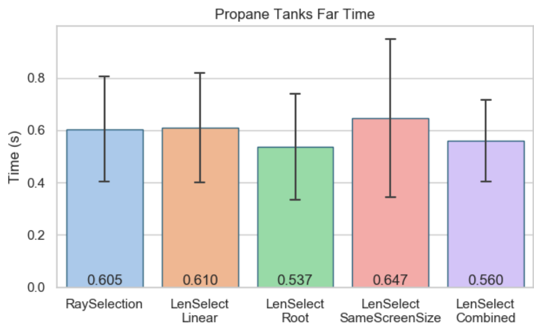


Figure 29.: Vive Prestudy: Average Index of Performance for the "Propane Tanks Far" test scenario.

With $(n = 70)$.
But we can also see that LenSelect Combined performs better under certain circumstances than LenSelect Root, for example the "Propane Tanks Far" test scenario.
The average ID for this test scenario is highly significant. $(F(4, 164.42) = 235.67, p < 0.01, \eta^2 = 0.59)$. Here LenSelect Combined $(M = 1.21, SD = 0.31)$ shows the lowest ID and significantly lower values than RaySelection $(M = 2.11, SD = 0.16)$, LenSelect Linear $(M = 1.51, SD = 0.19)$, LenSelect Root $(1.50, SD = 0.10)$ and LenSelect SameScreenSize $(M = 1.43, SD = 0.38)$. But also shows a higher standard deviation. While RaySelection performs significantly worse than all other selection techniques.
But this does not show much of an effect on selection times, as significant differences

can not be observed here by post-hoc tests, even though the Welch ANOVA is significant ($F(4, 170.95) = 2.50, p = 0.044$). RaySelection ($M = 0.60, SD = 0.20$), LenSelect Linear ($M = 0.61, SD = 0.21$), LenSelect Root ($M = 0.54, SD = 0.20$), LenSelect SameScreenSize ($M = 0.65, SD = 0.31$) and LenSelect Combined ($M = 0.16$).
The same goes for selection errors ($F(4, 138) = 2.95, p = 0.022$). RaySelection ($M = 0.07, SD = 0.26$), LenSelect Linear ($M = 0.06, SD = 0.29$), LenSelect Root ($M = 0.07, SD = 0.39$), LenSelect SameScreenSize ($M = 0.06, SD = 0.0$) and LenSelect Combined ($M = 0.0, SD = 0.0$). At the same time LenSelect Combined showed 0 standard deviation and therefore could not be included in the post-hoc test. So it probably is significant.
So depending on the use-case, different scaling methods might be preferable.

Results:
Hypothesis 1 came true, LenSelect Root shows a smaller ID and less selection errors than the other selection techniques. Even it's task completion time is better, although it is not clear if it is a noticeable difference. The low number of participants could be a factor here. Hypothesis 2 also came true LenSelect Root is definitely the preferred selection technique among users.
Also consider that sometimes the ID could not be calculated, this is the reason why sample sizes differ.

6.3.2. Powerwall

The study had n=7 participants, all of which self-identified as male and were either students or research associates. One person was left handed and none had a form of color blindness. Their ages ranged from 22 to 30 with a mean age of 27.57 and a standard deviation of 2.77 and a median age of 29.
Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	4.142	4	0.832
I'm often playing first person shooters.	2.857	3	1.124
I have experience with 3D-pointing devices.	4.142	4	0.638
I'm skilled at video games.	4	4	1.069
I have experience with HMD.	4	4	0.925

Table 3.: Powerwall Prestudy: Familiarity of participants with the presented tasks.

Participants show a rather high familiarity with the presented tasks.

Hypothesis 1: LenSelect Root will perform best of all selection techniques.
Hypothesis 2: LenSelect Root will be the preferred selection technique of participants.

6.3.2.1. Questionnaire Data

None of the results seem to be significant regarding these groups. The Kruskal-Wallis test shows no significant results for either Complexity ($H(4) = 6.69, p = 0.15$), Fun ($H(4) = 4.61, p = 0.33$), QUESI Total ($H(4) = 1.11, p = 0.89$) or User Ranking ($H(4) = 8.74, p = 0.07$).

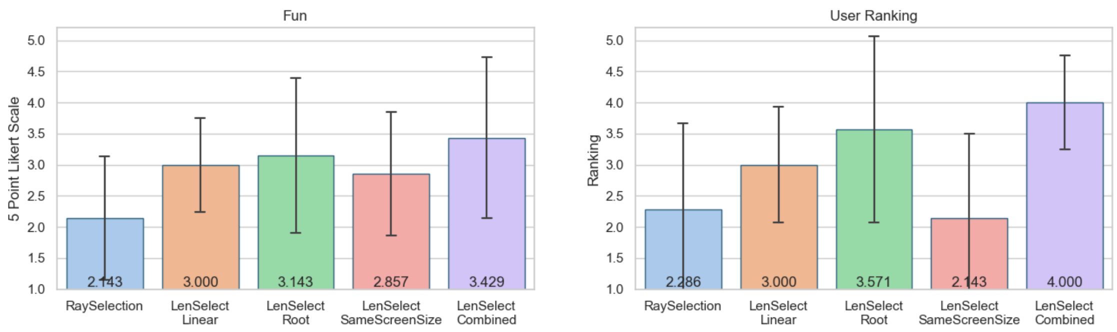


Figure 30.: Powerwall Prestudy: Fun rating and user ranking per selection technique. A higher value is better.

Results are similar to those of the Vive PreStudy, only they don't seem to be significant. SameScreenSize scaling still performs worst of all scaling functions. Still LenSelect Combined and LenSelect Root are the preferred selection techniques, even if not significantly so. Furthermore, this time, LenSelect Combined shows a slightly higher ranking than LenSelect Root.
Since equivalence analysis is inconclusive here, it is hard to say which selection technique is the best. But as LenSelect Root and Combined still share the best results and performed best in the previous study. It's probably safe to assume they performed the best here as well. Again no significant effect could be found for the QUESI scales.
Additionally one participant reported discomfort with the visual disruption from scaling the objects.

6.3.2.2. Fitts' Data

Again an outlier was removed for LenSelect Root, due to the participant being distracted.

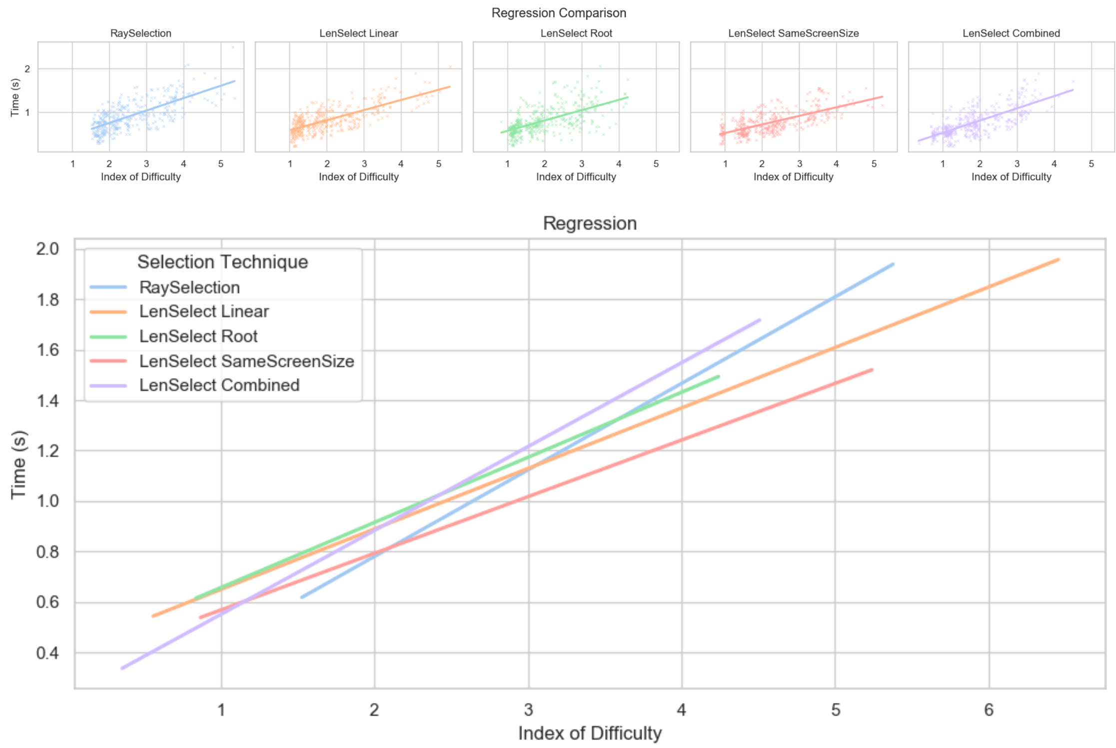


Figure 31.: Powerwall Prestudy: Regression Data for all selection techniques.

Again LenSelect techniques show a collection of data points at an ID RaySelection does not reach. Specifically LenSelect Combined produces a very low ID. LenSelect SameScreenSize starts producing better selection times at an ID ~ 2. Others take much longer before they produce better selection times than RaySelection. Weirdly LenSelect SameScreenSize seems to perform best according to this data. A notion which is disproved by the following results.

Average Index of Difficulty and Task Completion Time:

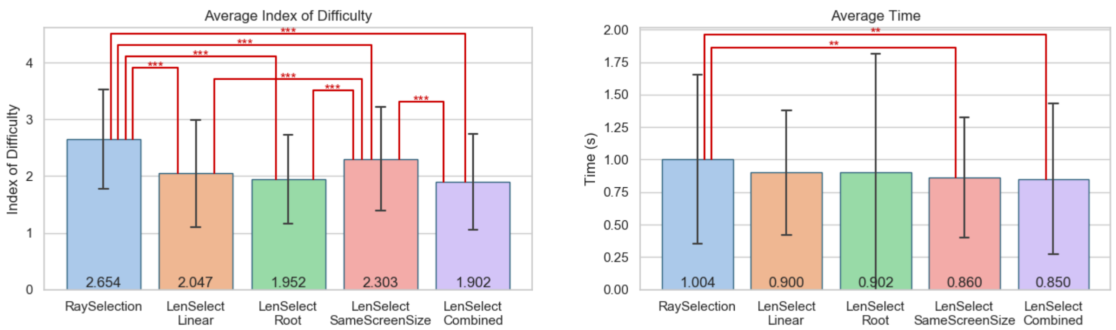


Figure 32.: Powerwall Prestudy: Average Index of Difficulty per selection technique.

With RaySelection ($n = 411$), LenSelect Linear ($n = 459$), LenSelect Root ($n =$

433), LenSelect SameScreenSize ($n = 435$) and LenSelect Combined ($n = 454$).
The average Index of Difficulty is highly significant ($F(4, 1090.45) = 54.83, p < 0.001, \eta^2 = 0.09$). Here RaySeelction ($M = 2.65, SD = 0.88$) performs the worst of all selection techniques and shows a significantly higher ID than the other techniques. LenSelect SameScreenSize ($M = 2.30, SD = 0.91$) also performs significantly worse than LenSelect Linear ($M = 2.05, SD = 0.95$), LenSelect Root ($M = 1.95, SD = 0.78$) and LenSelect Combined ($M = 1.90, SD = 0.48$). LenSelect Linear, LenSelect Root and LenSelect Combined seem to perform similar compared to each other.
This data is also significant ($F(4, 1074.9) = 4.18, p = 0.002, \eta^2 = 0.007$) for task completion time. Here RaySelection ($M = 1.0, SD = 0.65$) shows significantly higher selection times than LenSelect SameScreenSize ($M = 0.86, SD = 0.46$) and LenSelect Combined ($M = 0.85, SD = 0.58$). LenSelect Root ($M = 0.9, SD = 0.91$) is not significant due to an outlier, as can be seen in the high standard deviation. Removing this outlier leads to LenSelect Root also being significant in regards to RaySelection. So, only LenSelect Linear ($M = 0.9, SD = 0.48$) is not significant. The difference here is the same as in the Vive prestudy with $\sim 100ms$.

Average Selection Errors:

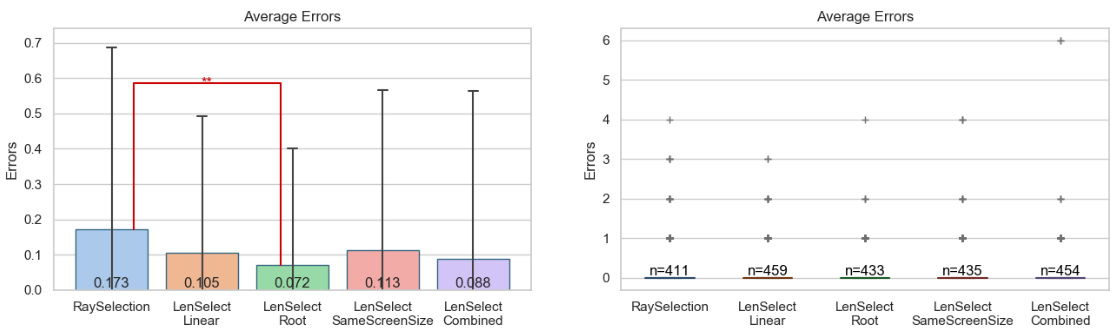


Figure 33.: Powerwall Prestudy: Average selection errors per selection technique.

The average selection error per selection technique is also significant ($F(4, 1081.28) = 3.01, p = 0.017$). LenSelect Root ($M = 0.07, SD = 0.33$) performs significantly better than RaySelection ($M = 0.17, SD = 0.51$). All other selection techniques show no significant differences. LenSelect Root is the only selection technique with a significant effect. It again shows about half the selection errors that RaySelection does. LenSelect Linear ($M = 0.10, SD = 0.39$), LenSelect SameScreenSize ($M = 0.11, SD = 0.45$).

Throughput TP_b:
Again RaySelection shows a lower intercept. But the slope is much closer to that of the LenSelect techniques. LenSelect SameScreenSize shows the lowest slope.

	RaySelection	LenSelect Linear	LenSelect Root	LenSelect Same-ScreenSize	LenSelect Combined
a	0.194	0.361	0.342	0.334	0.253
b	0.282	0.230	0.236	0.195	0.279
TP_b	3.540	4.340	4.240	5.122	3.587

Table 4.: Powerwall Prestudy: a, b and TP_b for different selection techniques.

LenSelect Root and Combined again seem to perform best of all selection techniques. Although the differences in selection times are less pronounced this time. This might be because of the lack of contrast on the Powerwall. As the lens is colored blue it might be harder to make out if the target object can be selected already or not.

Propane Tanks Far Index of Difficulty and Task Completion Time:

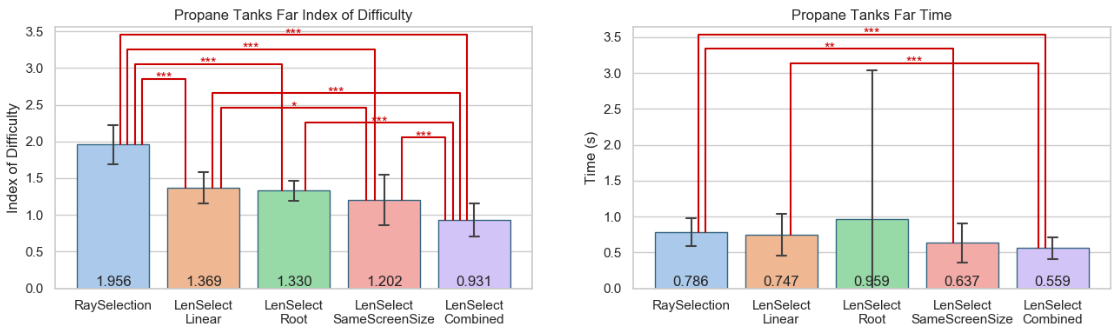


Figure 34.: Powerwall Prestudy: Index of Difficulty and task completion time for the "Propane Tanks Far" test scenario per selection technique.

With RaySelection ($n = 59$) and the other techniques $n = 61$.
LenSelect Combined again performs better than all other tested selection techniques in certain circumstances. As is the case with the ID for the "Propane Tanks Far" test scenario, which is highly significant ($F(4, 146.95) = 11.4, p < 0.001, \eta^2 = 0.156$). LenSelect Combined ($M = 0.93, SD = 0.23$) shows a significantly lower ID than all other selection techniques. While RaySelection ($M = 1.96, SD0.28$) shows a significantly higher ID than LenSelect Linear($M = 1.37, SD = 0.21$), LenSelect Root ($M = 1.33, SD = 0.14$), LenSelect SameScreenSize ($M = 1.20, SD = 0.34$) and LenSelect Combined. At the same time a higher standard deviation can be observed for LenSelect Combined compared to LenSelect Root.
This time the lowered ID also results in lower selection times, however. As task completion times are also highly significant ($F(4, 143.48) = 14.45, p < 0.001, \eta^2 = 0.020$). Here LenSelect Combined ($M = 0.56, SD = 0.15$) produces significantly lower selection times than RaySelection ($M = 0.79, SD = 0.20$) and LenSelect Linear ($M = 0.75, SD = 0.29$). Removing the very high outlier for LenSelect Root

($M = 0.96, SD = 2.10$) also reveals a significant effect between it and LenSelect Combined . At the same time LenSelect SameScreenSize ($M = 0.64, SD = 0.28$) produces significantly lower selection times compared to RaySelection.

Results:
It's not entirely clear if Hypothesis 1 and 2 came true, even though LenSelect Root and Combined perform slightly better than other selection techniques, their results are not always significant. At best one can say both Root and Combined scaling are equally preferred and performed equally well.
Still, LenSelect Root and LenSelect Combined prove to be the most promising of the tested selection techniques. And taking the results of the previous study into account both of them will be included in the final study.
Interestingly the ID seems generally lower for the Powerwall selections compared to those from the HTC Vive, this might be because the distance to objects does not perfectly match that of the Vive.

Conclusion:
Both LenSelect Root and LenSelect Combined will be used for the final studies with the Vive and Powerwall.

6.4. Best Parameter Prestudy

This study serves to find the best parameters for the given selection techniques from the first prestudy. As such LenSelect Root and LenSelect Combined will be evaluated here. The goal of this study is to find the right opening angle for the lens of both selection techniques and if said lens should be visible or not. There again will be separate studies for Vive and Powerwall.
The opening angles to evaluate are: Small = 10°, medium = 15°, big = 20°. It only uses five of the nine test scenarios and a reduced version of the QUESI questionnaire with only one question per scale. The removed selection techniques are: "Propane Tanks Close", "Erratic Spheres" "Fast Sphere" and "Rotating Cans".

6.4.1. Vive

The study had n=7 participants, all of which self-identified as male and were either students, research associates or employed in a related field. One person was left handed and none had a form of color blindness. Their ages ranged from 22 to 29 with a mean age of 23.85, with a standard deviation of 2.55 and a median age of 28. Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	3.714	4	1.385
I'm often playing first person shooters.	2.428	2	1.178
I have experience with 3D-pointing devices.	4.285	5	0.88
I'm skilled at video games.	4.428	5	1.049
I have experience with HMD.	4.714	5	0.451

Table 5.: Vive Parameter Study: Familiarity of participants with the presented tasks.

Participants show a high familiarity with the presented tasks.

Hypothesis 1: A medium-sized lens will be preferred.
Hypothesis 2: A visible lens will be preferred.

Opinions about the lens's size seem to differ. One participant thought the small cone is too small, another liked the small cone better than the rest. Even the lens's visibility is not as clear cut, one participant liked the visible cone for practice but found it distracting for prolonged use. Another thought a visible lens was better. It's hard to say if the parameters are very subjective or if the data is lacking.

6.4.1.1. Questionnaire Data Combined Scaling

Complexity and Fun:

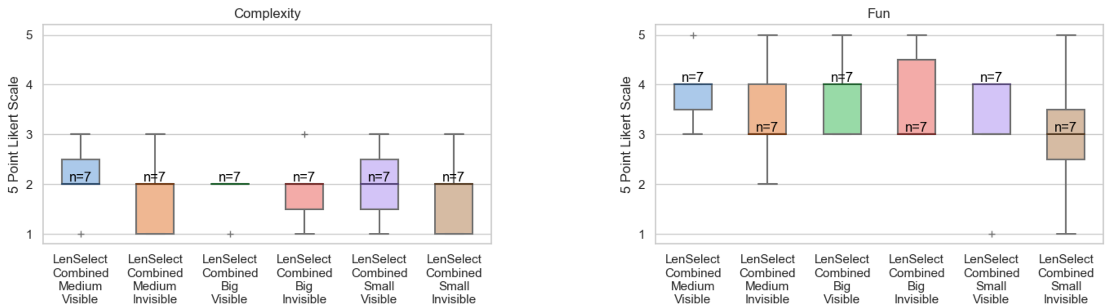


Figure 35.: Vive Parameter Study (Combined Scaling): Average complexity and fun rating per selection technique. A lower value is better for the complexity rating, while a higher value is better for the fun rating.

The result for the complexity rating is not significant ($H(5) = 2.19, p = 0.823$). This might be evidence that the complexity stays the same regardless of size and visibility of the used lens. This is supported by the medians of all selection techniques being the same.
The result for the fun rating is also not significant ($H(5) = 3.09, p = 0.687$). The same conclusions as with the complexity rating apply here. Only the medians seem to differ between visible and invisible lenses.

QUESI Total and User Ranking:

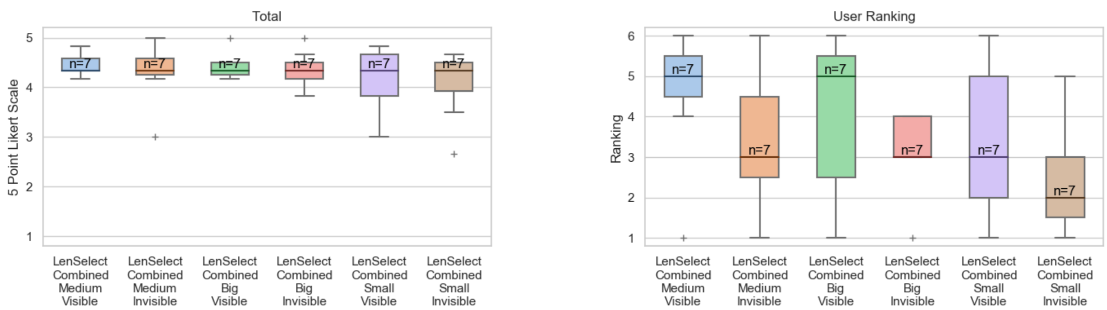


Figure 36.: Vive Parameter Study (Combined Scaling): Average QUESI Total and user ranking per selection technique. A higher value is better.

Even the QUESI Totals ($H(5) = 1.00, p = 0.962$) show no significant difference with the median staying roughly the same across lens parameters.
The QUESI Total isolated for lens visibility also shows no significant differences between visible and invisible lenses ($H(1) = 0.26, p = 0.608$).
As does the total isolated for lens size $H(2) = 0.72, p = 0.698$.

According to the Kruskal-Wallis test there is also no significant effect for the User Ranking ($H(5) = 6.39, p = 0.271$).
The user ranking isolated for lens visibility shows no significant effect either ($H(1) = 3.47, p = 0.062$). Even though there is a tendency for invisible lenses to score lower than visible ones.
Isolating for lens size also shows no effect ($H(2) = 2.85, p = 0.241$). Here a tendency for small lenses to score lower can be observed, with medium lenses having a tendency to perform slightly better than big lenses.

Results:
Means and medians are close together usually. So not many conclusions can be drawn from the questionnaire. Using a medium-sized, visible lens seems like the most agreeable course of action. As even though there is no clear preference among users, a slight tendency toward these parameters can be observed. But the opacity of the lens will be reduced to improve visibility of target objects. Both Hypothesis 1 and 2 came true.

6.4.1.2. Questionnaire Data Root Scaling

Complexity and Fun:

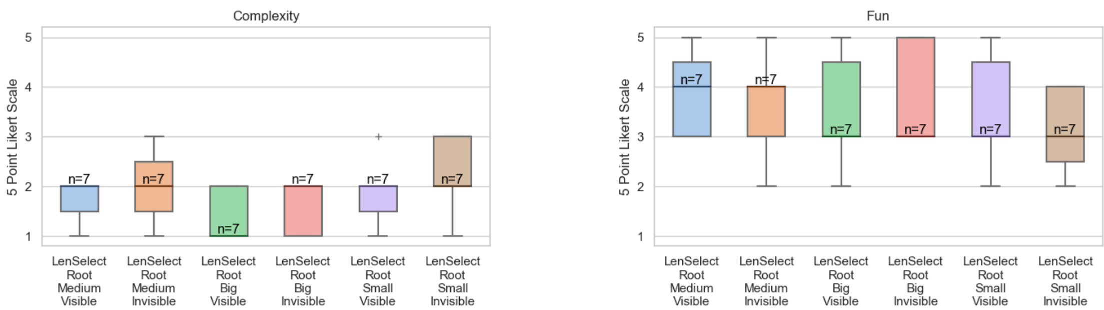


Figure 37.: Vive Parameter Study (Root Scaling): Average complexity and fun rating per selection technique. A lower value is better for the complexity rating, while a higher value is better for the fun rating.

For LenSelect Root no significant effect could be found for the complexity rating ($H(5) = 6.76, p = 0.239$) either. The same hypothesis as with combined scaling can be made here, that the complexity does not differ with different lens parameters.
The fun rating also showed no significant effect ($H(5) = 2.06, p = 0.841$). This time there seems to be a tendency for medium-sized lenses to perform better.

QUESI Total and User Ranking:
The QUESI Total shows no significant effect as well ($H(5) = 4.95, p = 0.422$).
Isolating for visibility ($H(1) = 1.14, p = 0.285$) or lens size ($H(2) = 2.49, p = 0.288$) also shows no significant effect.

There is also no significant effect for the user ranking ($H(5) = 5.59, p = 0.348$). Isolating the user ranking for lens visibility also yields no significant effect ($H(1) = 2.58, p = 0.108$). Although, again, a tendency towards a visible lens can be observed. Isolating for lens size also shows no significant effect ($H(2) = 2.97, p = 0.227$). Again a tendency of the small lens scoring lower can be observed.

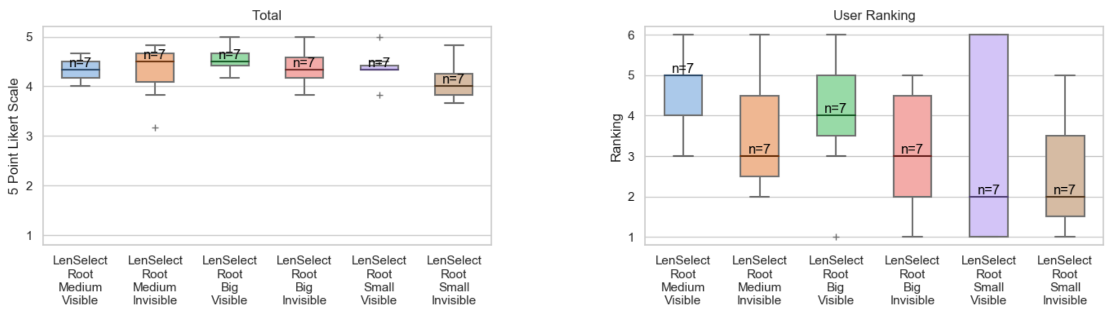


Figure 38.: Vive Parameter Study (Root Scaling): Average QUESI Total and user ranking per selection technique. A higher value is better.

Results: The results are practically identical to those of LenSelect Combined, a visible, medium-sized lens seems to be preferred by participants. This means both Hypothesis 1 and 2 came true.

6.4.1.3. Fitts' Data Combined Scaling

The Welch ANOVA shows no significant effect for the Index of Difficulty ($F(5, 483.88) = 0.078, p = 1.0, \eta^2 < 0.001$). The differences between lens parameters are also minuscule. This makes sense as a bigger or smaller lens should not have any bearing on the Index of Difficulty, as should a visible lens, unless it makes it harder to see the objects.
Average task completion time also shows no significant effect ($F(5, 483.45) = 0.65, \eta^2 = 0.003$).
As does average errors ($F(5, 479.76) = 0.29, p = 0.92, \eta^2 = 0.001$). In fact, all of these are almost identical. The same effect can be observed in intercept and slope for these different lens parameters.

	Medium Visible	Medium Invisible	Big Visible	Big Invisible	Small Visible	Small Invisible
a	0.338	0.331	0.246	0.303	0.310	0.333
b	0.146	0.164	0.173	0.169	0.159	0.154
TP_b	6.843	6.089	5.778	5.914	6.259	6.467

Table 6.: Vive Parameter Study (Combined Scaling): a, b and TP_b with different lens parameters.

6.4.1.4. Fitts' Data Root Scaling

The same observations as for Combined Scaling apply to Root Scaling.
The average ID shows no significant effect ($F(5, 481.36) = 0.04, p = 1.0, \eta^2 < 0.001$). The same goes for average task completion time ($F(5, 480.52) = 1.61, p = 0.156, \eta^2 = 0.007$) and average selection errors ($F(5, 480.31) = 0.56, p = 0.727, \eta^2 = 0.004$).
In the end Fitts' data does not seem useful for deciding which lens parameters to choose. Just as before the results are almost identical for all of these scales. And intercept and slope are similar as well.

	Medium Visible	Medium Invisible	Big Visible	Big Invisible	Small Visible	Small Invisible
a	0.355	0.301	0.234	0.261	0.396	0.354
b	0.170	0.196	0.215	0.202	0.134	0.142
TP_b	5.865	5.086	4.647	4.939	7.455	7.038

Table 7.: Vive Parameter Study (Root Scaling): a, b and TP_b with different lens parameters.

6.4.2. Powerwall

The same conditions apply as with the Vive parameter study, but the Powerwall test environment as described in Chapter 6.3.2 Powerwall on p. 59 is used.
The study had n=7 participants, all of which self-identified as male and were either students, research associates or employed in a related field. Two people were left handed and none had a form of color blindness. Their ages ranged from 22 to 29 with a mean age of 25.85, a standard deviation of 2.76 and a median age of 26.
Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	4	4	0.925
I'm often playing first person shooters.	2.714	2	1.277
I have experience with 3D-pointing devices.	4.428	4	1.049
I'm skilled at video games.	3.857	4	1.245
I have experience with HMD.	4.142	5	1.355

Table 8.: Powerwall Parameter Study: Familiarity of participants with the presented tasks.

Participants show to be familiar with the presented tasks.

Hypothesis 1: A medium sized lens will be preferred by participants.
Hypothesis 2: A visible lens will be preferred by participants.

6.4.2.1. Questionnaire Data Combined Scaling

Complexity and Fun:

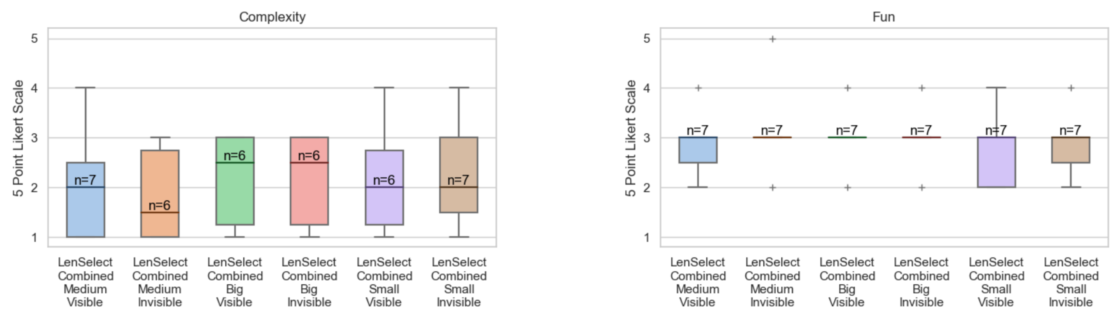


Figure 39.: Powerwall Parameter Study (Combined Scaling): Average complexity and fun rating per selection technique. A lower value is better for complexity, while a higher value is better for fun.

The complexity rating is not significant again, according to the Kruskal-Wallis test ($H(5) = 0.82, p = 0.976$). Again the assumption that complexity is independent of lens parameters might be made.
No significant effect could be found for the fun rating either ($H(5) = 1.47, p = 0.916$). This time the dispersion of the data is almost non-existent, this is further evidence that the fun rating is independent of lens size and visibility, to a certain degree.

QUESI Total and User Ranking:

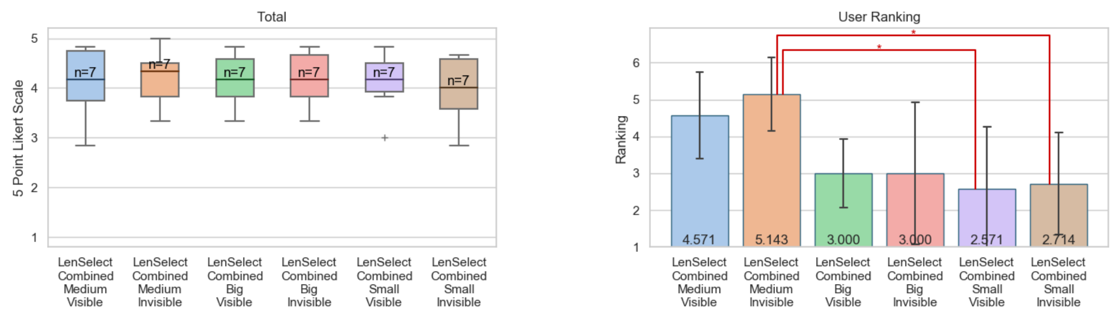


Figure 40.: Powerwall Parameter Study (Combined Scaling): Average QUESI Total and user ranking per selection technique. A higher value is better.

The QUESI Total shows no significant effect, as well ($H(5) = 0.52, p = 0.99$). Like the fun rating the dispersion is rather low.
Isolating for visibility ($H(1) = 0.04, p = 0.85$) and lens size ($H(2) = 0.37, p = 0.833$)

also shows no significant effect.
The user ranking, however, does show a significant effect ($H(5) = 13.65, p = 0.018$). Where the medium, invisible lens ($M = 5.14, SD = 1.07$) is rated significantly higher than both of the small, visible ($M = 2.57, SD = 1.81$) and invisible lens ($M = 2.71, SD = 1.50$). No significant effects could be observed for the other lenses, but the medium, visible lens ($M = 4.57, SD = 1.27$) is almost rated equally as high as the other medium-sized lens. Big, visible lens ($M = 3.0, SD = 1.0$), big, invisible lens ($M = 3.0, SD = 2.08$).

User Ranking Isolated for Lens Size and Visibility:

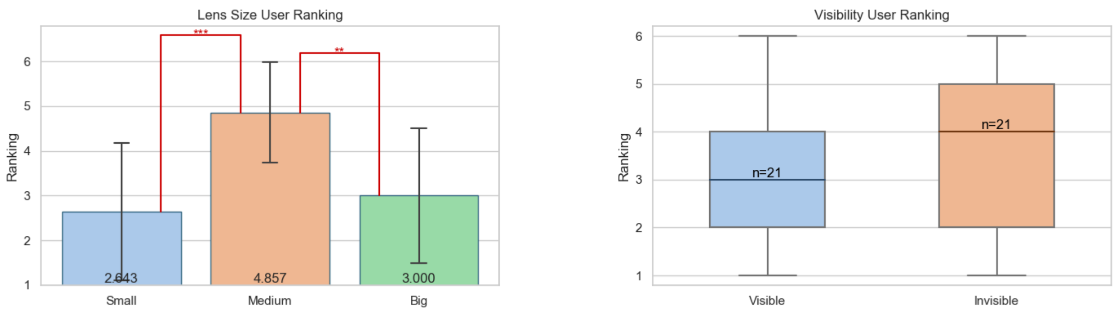


Figure 41.: Powerwall Parameter Study (Combined Scaling): User ranking isolated for lens size and visibility per selection technique. A higher value is better.

Meanwhile isolating for lens size revealed a significant effect ($H(2) = 13.24, p = 0.001$). Where the medium lens ($M = 4.15, SD = 0.63$) was rated significantly higher than the small ($M = 4.05, SD = 0.63$) and the big lens ($M = 4.18, SD = 0.54$).
Isolating the ranking for lens visibility yielded no significant effect however ($H(1) = 0.20, p = 0.655$). But a small tendency towards an invisible lens can be observed.

Results:
A medium lens is clearly preferred by users. Data is inconclusive for lens visibility, however. So hypothesis 1 came true, while hypothesis 2 is unclear.

6.4.2.2. Questionnaire Data Root Scaling

Complexity and Fun:
No significant effect could be found for the complexity rating ($H(5) = 3.58, p = 0.612$). Like before dispersion is minimal.
The same goes for the fun rating ($H(5) = 5.01, p = 0.414$). As in the pages before, results differ only slightly.

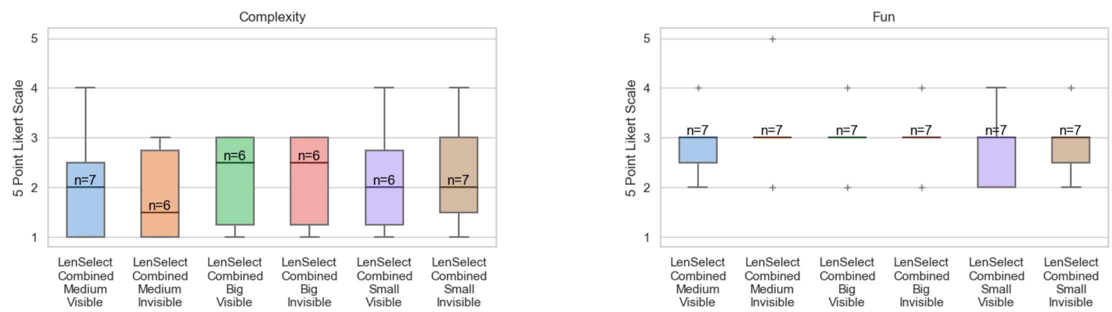


Figure 42.: Powerwall Parameter Study (Combined Scaling): Complexity and fun rating. A lower value is better for the complexity rating, while a higher value is better for the fun rating.

QUESI Total:
Neither the QUESI Total ($H(5) = 2.54, p = 0.770$) nor isolating for either visibility ($H(1) = 0.01, p = 0.929$) or lens size ($H(2) = 1.07, p = 0.585$) shows a significant effect. Results shows only minor differences again.

User Ranking:

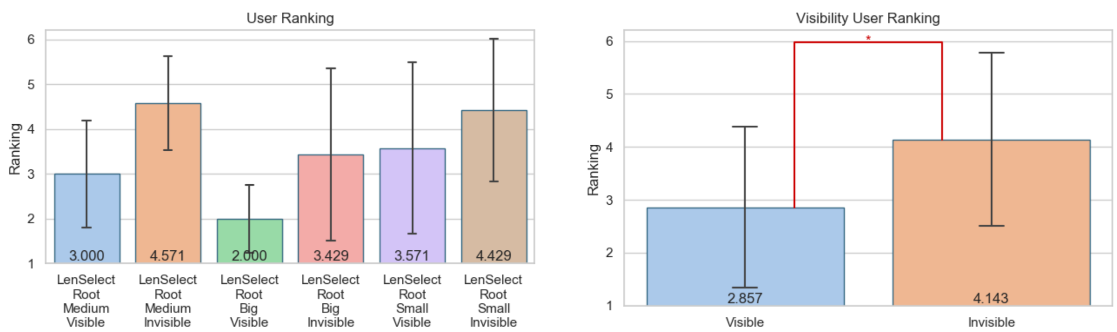


Figure 43.: Powerwall Parameter Study (Root Scaling): Average user ranking, a higher value is better.

The User Ranking also shows no significant effect ($H(5) = 10.59, p = 0.060$). Even though results differ much more than before. The p-value is almost significant even. But isolating for lens visibility does ($H(1) = 5.81, p = 0.016$). Where the invisible ($M = 4.14, SD = 1.68$) lens is significantly preferred over the visible one ($M = 2.86, SD = 1.56$).
Isolating for lens size does not show any effect on the other hand ($H(2) = 4.45, p = 0.108$). Again the p-value comes close to being significant and at least the big lens seems to be the least preferred.

Results:
Combining the findings of LenSelect Root and LenSelect Combined shows a lens

that is medium and invisible seems to be preferred. The sample size for this data is rather small, this might be an explanation as to why the results seem to differ between selection techniques. Hypothesis 1 did come true, a medium-sized lens is preferred. But hypothesis 2 did not, an invisible lens is preferred.

6.4.2.3. Fitts' Data Combined Scaling

No significant effect was found for the average Index of Difficulty ($F(5, 486.71) = 0.05, p = 0.998, \eta^2 < 0.001$).
The Welch ANOVA also shows no significant effect for average task completion time ($F(5, 483.85) = 1.63, p = 0.150, \eta^2 = 0.006$) or average selection errors ($F(5, 484.39) = 0.90, p = 0.482, \eta^2 = 0.004$).
Again only minor differences could be found for ID, task completion time, selection errors, slope and intercept.

	Medium Visible	Medium Invisible	Big Visible	Big Invisible	Small Visible	Small Invisible
a	0.315	0.387	0.309	0.360	0.444	0.397
b	0.284	0.255	0.263	0.263	0.189	0.228
TP_b	3.514	3.914	3.797	3.796	5,279	4.383

Table 9.: Powerwall Parameter Study (Combined Scaling): a, b and TP_b for different lens parameters.

6.4.2.4. Fitts' Data Root Scaling

Again no significant effect could be found for the average Index of Difficulty ($F(5, 487.16) = 0.07, p = 0.996, \eta^2 < 0.001$). As well as average task completion time ($F(5, 486.69) = 0.82, p = 0.536, \eta^2 = 0.004$) or average errors ($F(5, 485.87) = 0.29, p = 0.920, \eta^2 = 0.001$).
As already remarked earlier, Fitts' data is not suited for this comparison and will be disregarded for this study. As clearly the results show only minor differences again.

	Medium Visible	Medium Invisible	Big Visible	Big Invisible	Small Visible	Small Invisible
a	0.390	0.503	0.357	0.464	0.373	0.466
b	0.268	0.186	0.270	0.221	0.253	0.205
TP_b	3.730	5.359	3.701	4.514	3.949	4.855

Table 10.: Powerwall Parameter Study (Root Scaling): a, b and TP_b for different lens parameters.

Conclusion:
There isn't much conclusive data to support a decision. However it seems an invisible, medium-sized lens is preferred by participants for selection on the Powerwall. Results are combined for both LenSelect techniques for better comparability and because both scaling factors did not show much of a difference between each other. Therefore it is assumed both show the same response from participants. A decision is also harder to make due to the low number of study participants.

6.5. Final Study

In this study LenSelect Root and LenSelect Combined will be compared to IntenSelect and Expand, RaySelection will again serve as a baseline. The parameters gathered from the best parameter study will be applied to LenSelect Root and LenSelect Combined respectively.
After some testing the parameters for IntenSelect were chosen to be: 0.8 for the Stickiness, 0.2 for the Snappiness and 0.9 for k.

Hypothesis 1: IntenSelect will be the most accepted selection technique.
Hypothesis 2: Expand will have the longest selection times.

6.5.1. Vive

The study had n = 20 participants, 15 of which self-identified as male and 5 as female. Most of them were either research- or scientific assistants or students. All of the participants were familiar in the field of Computer Science. Four people were left handed and none had a form of color blindness. Their ages ranged from 20 to 35 with a mean age of 26.71, a standard deviation of 4.30 and a median of 27.5.
Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	3.8	4	0.872
I'm often playing first person shooters.	2.35	2	1.014
I have experience with 3D-pointing devices.	4.1	4	0.831
I'm skilled at video games.	3.75	4	0.942
I have experience with HMD.	3.75	4	1.260

Table 11.: Vive Study: Familiarity of participants with the presented tasks.

Participants showed a familiarity with the presented tasks.
Due to Covid-19 and the subsequent quarantine finding enough participants was difficult. So some of the participants from previous studies also participated in the final study. This applies to a total of 7 participants.

6.5.1.1. Questionnaire Data

Complexity and Fun:
According to the Kurskal-Wallis test a significant effect can be observed in the complexity rating ($H(4) = 30.63, p < 0.001$). The Tukey post-hoc test reveals that Expand ($M = 3.35, SD = 1.14$) is rated significantly worse than all other selection techniques. And that 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 ($M = 1.95, SD = 0.83$). Both LenSelect techniques perform well when compared to RaySelection.
The fun rating also shows a significant effect ($H(4) = 14.70, p = 0.005$). 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 among themselves.

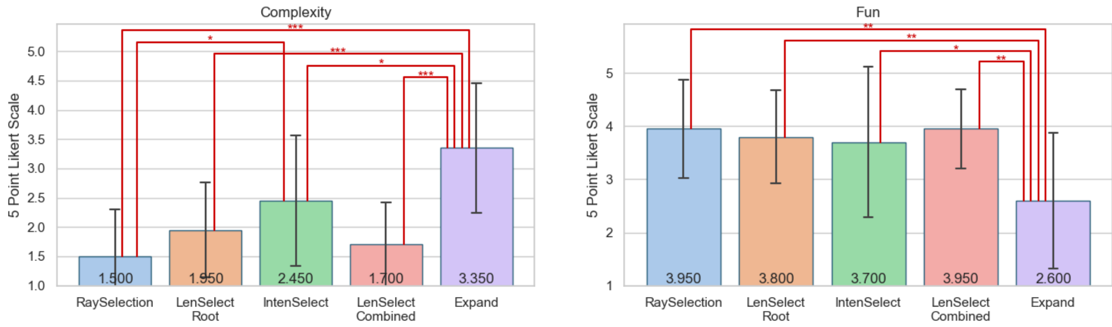


Figure 44.: Vive Study: Average complexity and fun rating per selection technique. A lower value is better for the complexity rating, while a higher value is better for the fun rating.

QUESI Total and User Ranking:

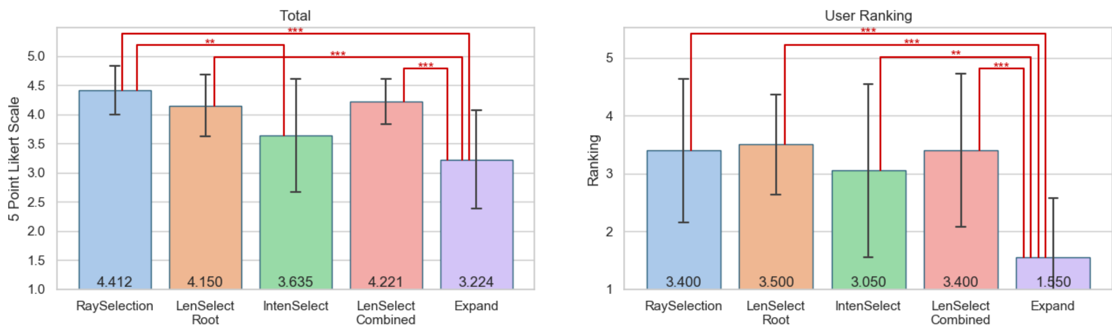


Figure 45.: Vive Study: QUESI Total and user ranking per selection technique. A higher value is better.

The QUESI Total is significant as well ($H(4) = 29.14, p < 0.001$). 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 ($M = 4.22, SD = 0.40$) as well. Both LenSelect techniques are rated

slightly lower than RaySelection, but not significantly so.
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 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 are about equally popular. LenSelect Root shows the lowest dispersion among it's data after RaySelection. The other techniques are more controversial.

Cognitive Load and Familiarity:

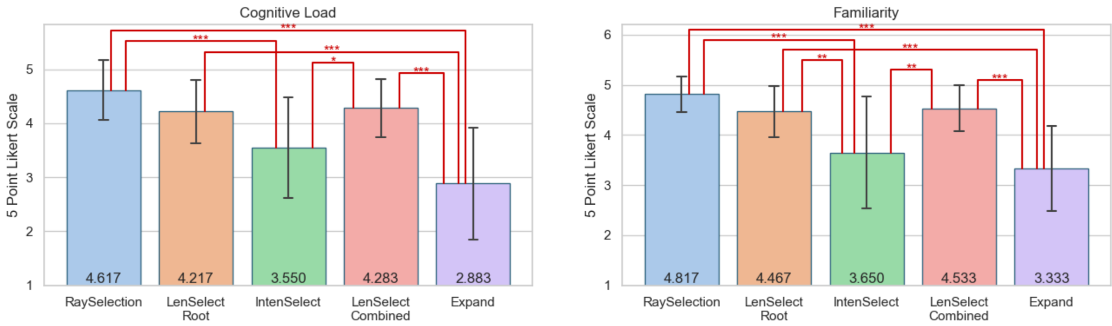


Figure 46.: Vive Study: Cognitive Load and Familiarity per selection technique. A higher value is better.

The Cognitive Load also shows a significant effect ($H(4) = 36.65, p < 0.001$). Expand ($M = 2.88, SD = 1.07$) performs significantly worse than RaySelection ($M = 4.62, SD = 0.56$), LenSelect Root ($M = 4.22, SD = 0.60$) and LenSelect Combined ($M = 4.28, SD = 0.55$). Only IntenSelect shows no significant difference compared to Expand. While IntenSelect ($M = 3.55, SD = 0.96$) performs significantly worse than RaySelection and LenSelect Combined. RaySelection is still the best according to this scale, followed by LenSelect Root and LenSelect Combined.
The Familiarity scale is significant, too ($H(4) = 40.48, p < 0.001$). Where RaySelection ($M = 4.82, SD = 0.37$) is rated highest and significantly higher than IntenSelect ($M = 3.65, SD = 1.15$) and Expand ($M = 3.33, SD = 0.88$), which is rated lowest. LenSelect Root ($M = 4.47, SD = 0.52$) and LenSelect Combined ($M = 4.53, SD = 0.48$) perform significantly better than IntenSelect and Expand.

Perceived Learning Effort:
The Perceived Learning Effort scale is also significant ($H(4) = 42.96, p < 0.001$). Expand ($M = 3.13, SD = 1.09$), again, performs worst and significantly so compared to RaySelection ($M = 4.90, SD = 0.19$), LenSelect Root ($M = 4.55, SD = 0.52$) and LenSelect Combined ($M = 4.55, SD = 0.52$). IntenSelect ($M = 3.77, SD = 1.14$) performs significantly worse than both LenSelect Root and LenSelect Combined. The dispersion for RaySelection is very low as well, only the LenSelect techniques

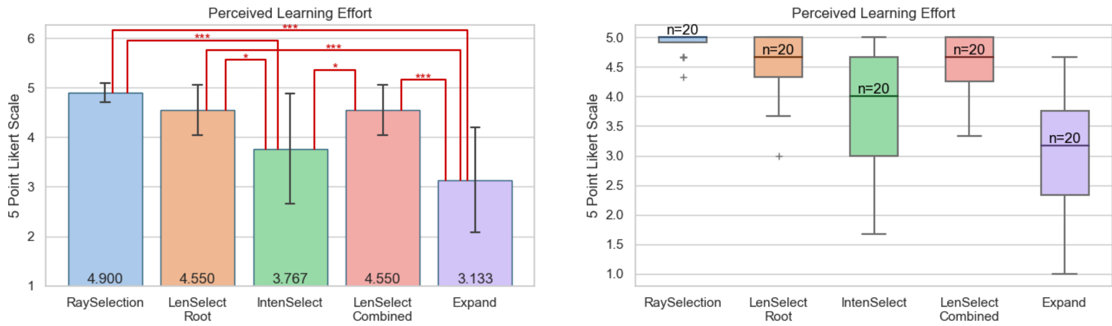


Figure 47.: Vive Study: Perceived Learning Effort per selection technique. A higher value is better.

come close. IntenSelect and Expand, on the other hand, show very high dispersion.

Results:
As can be seen Expand scores lowest on all scales in the questionnaire. It's the least intuitive and least popular selection technique. Both LenSelect techniques are rated higher or about the same as IntenSelect. LenSelect performs on par with RaySelection and is among the most intuitive selection techniques. On all other QUESI scales no significant effects could be found.

6.5.1.2. Fitts' Data

Five samples had to be removed due to questions or distractions. Two for Expand, two for LenSelect Combined and one for IntenSelect.

	RaySelection	LenSelect Root	IntenSelect	LenSelect LenSelect Combined	Expand
a	0.207	0.421	0.380	0.401	1.562
b	0.247	0.185	0.257	0.182	0.266
TP_b	4.048	5.416	3.892	5.484	3.757

Table 12.: Vive Study: a, b and TP_b for different selection techniques.

LenSelect and IntenSelect produce low IDs, while Expand shows much higher IDs than the rest of the selection techniques. Expand also shows much higher selection times than the other techniques. LenSelect produces lower selection times than RaySelection at an ID of ~ 3. IntenSelect shows a low intercept but a higher slope than both LenSelect techniques. In practice this means LenSelect will perform better than IntenSelect with higher ID. As in the prestudies RaySelection shows the lowest intercept. But again the data is not completely comparable, as Inten- and LenSelect will produce a lower ID for the same selection due to their target scaling compared to RaySelection.

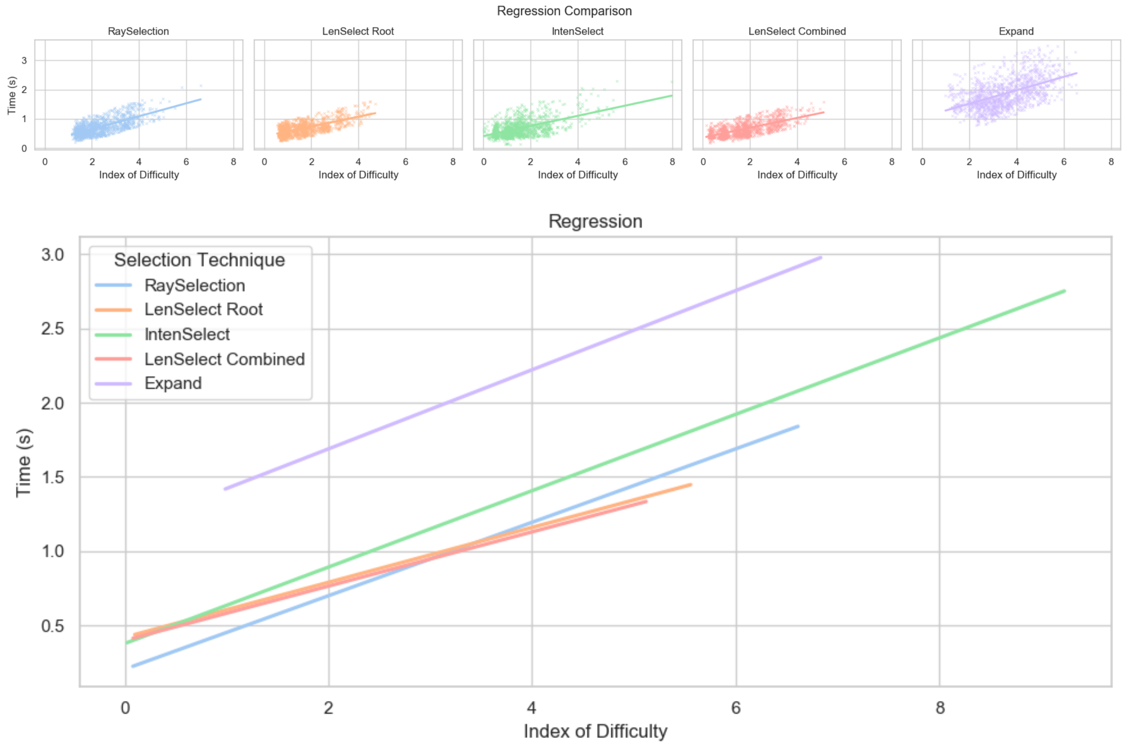


Figure 48.: Vive Study: Regression Data for all selection techniques.

Again, consider that samples outside of 95% the standard deviation of selection time around the regression were removed for the scatter plots shown here.

Average Index of Difficulty and Task Completion Time:

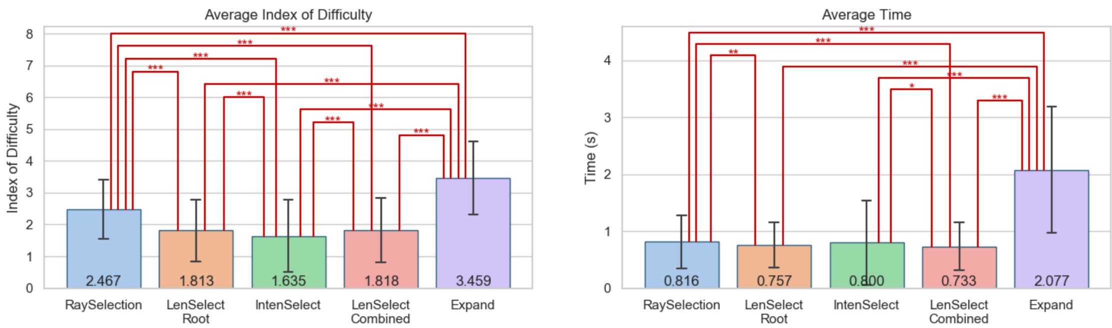


Figure 49.: Vive Study: Average Index of Difficulty and task completion time per selection technique.

With RaySelection ($n = 1496$), LenSelect Root ($n = 1499$), IntenSelect ($n = 1489$), LenSelect Combined ($n = 1495$) and Expand ($n = 1446$).
The Average Index of Difficulty is highly significant according to the Welch ANOVA ($F(4, 3698.82) = 668.34, p < 0.001, \eta^2 = 0.291$). According to the Games-Howell

post-hoc test almost all groups are significant compared to each other. With Expand ($M = 3.46, SD = 1.15$) performing the worst, followed by RaySelection ($M = 2.47, SD = 0.93$). Only LenSelect Root ($M = 1.81, SD = 0.97$) and LenSelect Combined ($M = 1.82, SD = 1.01$) are not significant compared to each other. While IntenSelect ($M = 1.84, SD = 1.13$) performs best of all selection techniques. Even if only slightly compared to LenSelect (~ 0.2).
The task completion time also shows a significant effect ($F(4, 3624.75) = 490.05, p < 0.001, \eta^2 = 0.366$). Expand ($M = 2.08, SD = 1.11$) performs the worst compared to all other selection techniques. LenSelect Root ($M = 0.76, SD = 0.34$) and LenSelect Combined ($M = 0.73, SD = 0.42$) perform significantly better than RaySelection ($M = 0.82, SD = 0.47$). While IntenSelect ($M = 0.80, SD = 0.74$) performs significantly worse than LenSelect Combined, but only marginally ($\sim 70ms$). LenSelect Root and IntenSelect perform on par with each other. And IntenSelect is not significant compared to RaySelection. IntenSelect does not manage to produce lower selection times, despite it's lower ID.

Average Selection Errors:

Figure 50.: Vive Study: Average selection errors per selection technique.

This scale shows a significant effect as well ($F(4, 3686.90) = 6.38, p < 0.001, \eta^2 = 0.004$). RaySelection ($0.20, SD = 0.55$) performs significantly worse than LenSelect Root ($M = 0.11, SD = 0.40$), IntenSelect ($M = 0.13, SD = 0.59$) and LenSelect Combined ($M = 0.13, SD = 0.42$). Expand ($M = 0.15, SD = 0.49$) shows no significant effect when compared to any other selection technique. Both LenSelect techniques and IntenSelect show a similar amount of selection errors. Expand shows a slightly higher amount of errors.

However results can differ with the test scenario.

Erratic Spheres:
With $n = 100$)
The selection errors for the "Erratic Spheres" test scenario show a significant effect ($F(4, 245.71) = 3.54, p = 0.008, \eta^2 = 0.033$). Here Expand ($M = 0.12, SD =$

0.43) and IntenSelect ($M = 0.08, SD = 0.37$) perform significantly better than RaySelection ($M = 0.34, SD = 0.65$) and the best out of all selection techniques. LenSelect Root ($M = 0.16, SD = 0.49$) and LenSelect Combined ($M = 0.21, SD = 0.43$) show no significant effect, even though they still show less selection errors than RaySelection. IntenSelect shows the least selection errors.

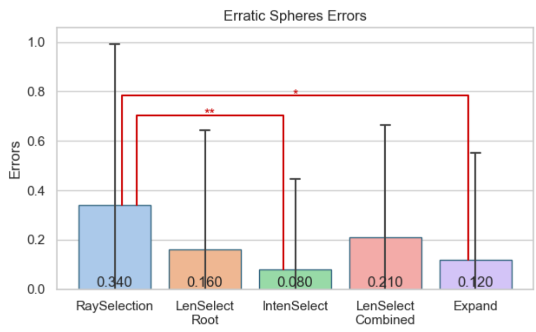


Figure 51.: Vive Study: Average selection errors for the “Erratic Spheres” test scenario per selection technique.

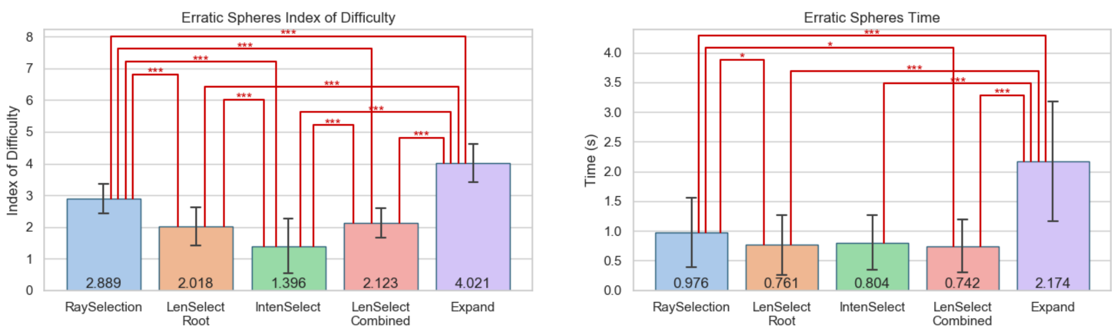


Figure 52.: Vive Study: Average Index of Difficulty and task completion time for the “Erratic Spheres” test scenario per selection technique.

For the ID a significant effect was observed ($F(4, 245.24) = 244.26, p < 0.001, \eta^2 = 0.68$) for the “Erratic Spheres” test scenario. Again all selection techniques are significant compared to each other except LenSelect Root ($M = 2.02, SD = 0.60$) and LenSelect Combined ($M = 2.12, SD = 0.47$), which are not significant compared to each other. Expand performs the worst ($M = 4.02, SD = 0.60$), while RaySelection ($M = 2.89, SD = 0.47$) performs better than Expand but worse than the rest of the selection techniques. IntenSelect ($M = 1.40, SD = 0.88$) performs best, followed by both LenSelect techniques.
The task completion time also shows a significant effect ($F(4, 244.35) = 45.35, p < 0.001, \eta^2 = 0.425$) for this test scenario. Here Expand ($M = 2.17, SD = 1.01$) takes significantly longer for a selection than all other techniques. While LenSelect Root ($M = 0.76, SD = 0.51$) and LenSelect Combined ($M = 0.74, SD = 0.45$)

are significantly faster than RaySelection ($M = 0.98, SD = 0.59$). IntenSelect ($M = 0.80, SD = 0.46$) only shows a significant effect compared to Expand. Which is peculiar, as IntenSelect performed better than all other techniques in terms of Index of Difficulty and selection errors. LenSelect Root and LenSelect Combined show the lowest selection times, but the difference is only minor $\sim 50ms$ compared to IntenSelect.

Fast Sphere:

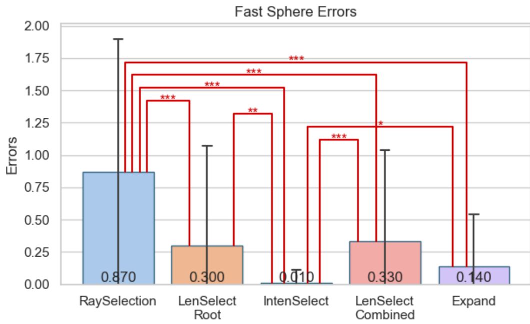


Figure 53.: Vive Study: Average errors for the “Fast Sphere” test scenario per selection technique.

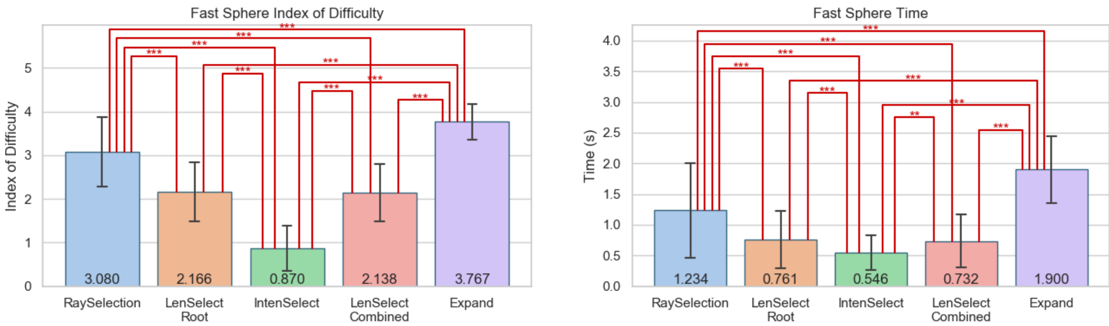


Figure 54.: Vive Study: Average Index of Difficulty and task completion time for the “Fast Sphere” test scenario per selection technique.

With $n = 100$.
For the “Fast Sphere” test scenario a significant effect can be observed for the selection errors ($F(4, 207.40) = 26.64, p < 0.001, \eta^2 = 0.157$). Here IntenSelect ($M = 0.01, SD = 0.10$) performs significantly better than all other selection techniques. RaySelection ($M = 0.87, SD = 1.03$) shows significantly more selection errors than all other selection techniques. LenSelect Root ($M = 0.30, SD = 0.77$), LenSelect Combined ($M = 0.33, SD = 0.71$) and Expand ($M = 0.14, SD = 0.40$) show no significant effect between each other, only compared to the aforementioned selection techniques. It’s of note here that IntenSelect shows almost no errors out

of 100 selections.
The Index of Difficulty is significant for the “Fast Sphere” test scenario ($F(4, 243.49) = 505.54, p < 0.001, \eta^2 = 0.710$). Here IntenSelect ($M = 0.87, SD = 0.52$) shows a significantly lower ID than all other selection techniques. While LenSelect Root ($M = 2.17, SD = 0.68$) and LenSelect Combined ($M = 2.14, SD = 0.70$) produce significantly lower IDs than RaySelection ($M = 3.10, SD = 0.80$) and Expand ($M = 3.77, SD = 0.41$). RaySelection still produces a significantly lower ID than Expand.
For the task completion time we can see a significant effect, as well ($F(4, 240.49) = 127.63, p < 0.001, \eta^2 = 0.466$). Again all selection techniques are significant compared to each other, except for LenSelect Root ($M = 0.761, SD = 0.47$) and LenSelect Combined ($M = 0.73, SD = 0.43$) which show no such effect between each other. IntenSelect ($M = 0.55, SD = 0.28$) performs the best for this test scenario. Expand ($M = 1.90, SD = 0.55$) shows the longest selection times, even if it minimizes selection errors. RaySelection ($M = 1.23, SD = 0.77$) takes the second longest. Here IntenSelect’s lower ID does lead to a faster task completion time. Overall the ID fits well with the task completion time for this test scenario, but this is not necessarily the case with all test scenarios

Propane Tanks Close:

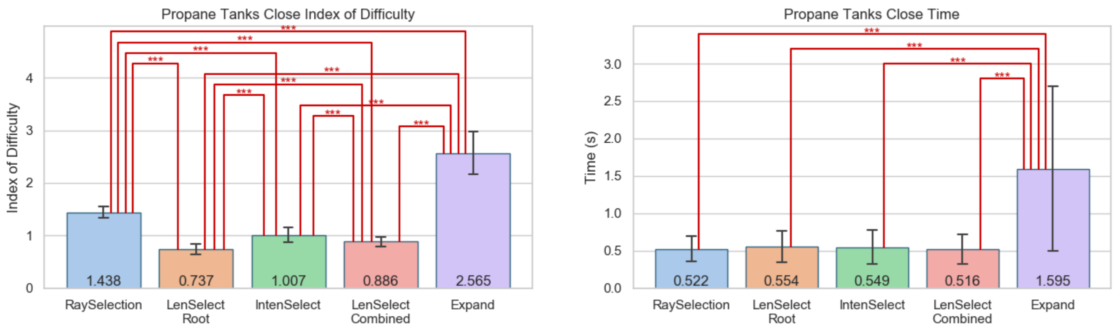


Figure 55.: Vive Study: Average Index of Difficulty and task completion time for the “Propane Tanks Close” test scenario per selection technique.

Here we can see that a lower ID might not necessarily lead to lower selection times. With RaySelection ($n = 198$), LenSelect Root ($n = 199$), IntenSelect ($n = 200$), LenSelect Combined ($n = 198$) and Expand ($n = 197$).
A significant effect ($F(4, 483.92) = 1994.42, p < 0.001, \eta^2 = 0.911$) can be observed for the ID of the “Propane Tanks Close” test scenario. Here LenSelect Root ($M = 0.74, SD = 0.10$) and LenSelect Combined ($M = 0.89, SD = 0.09,$) perform significantly better than all other selection techniques, except between themselves. IntenSelect ($M = 1.01, SD = 0.14$) comes a close second, performing better than RaySelection ($M = 1.44, SD = 0.10$) and Expand ($M = 2.57, SD = 0.41$). Expand, again, performs worst of all selection techniques.

None of the effects from the ID can be observed for the task completion time, however. A significant effect can be observed ($F(4, 484.23) = 46.26, p < 0.001, \eta^2 = 0.395$). But it only shows that Expand ($M = 1.60, SD = 1.11$) needs significantly more time than all other selection techniques. RaySelection ($M = 0.52, SD = 0.17$), LenSelect Root ($M = 0.55, SD = 0.21$), IntenSelect ($M = 0.55, SD = 0.27$) and LenSelect Combined ($M = 0.52, SD = 0.17$) show no significant difference between each other. Again a better ID fails to produce better selection times.

Propane Tanks Far:

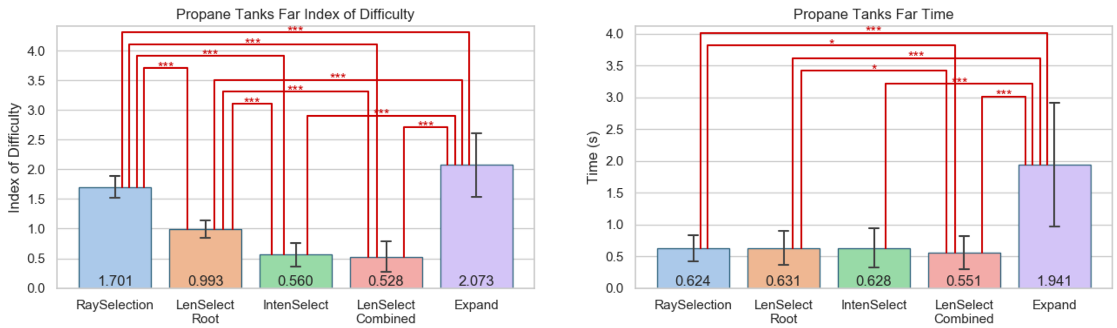


Figure 56.: Vive Study: Average Index of Difficulty and task completion time for the "Propane Tanks Far" test scenario per selection technique.

With RaySelection ($n = 200$), LenSelect Root ($n = 200$), IntenSelect ($n = 198$), LenSelect Combined ($n = 199$) and Expand ($n = 200$).
A significant effect can be observed ($F(4, 486.11) = 1311.67, p < 0.001, \eta^2 = 0.811$) for the "Propane Tanks Far" test scenario. Here Expand ($M = 2.07, SD = 0.53$) performs significantly worse than all other selection techniques. RaySelection ($M = 1.70, SD = 0.19$) follows, performing significantly worse than the rest of the selection techniques. LenSelect Combined ($M = 0.53, SD = 0.26$) shows the lowest ID and significantly lower than all other selection techniques. Followed by IntenSelect ($M = 0.56, SD = 0.19$) which only performs worse than LenSelect Combined and better than the rest. LenSelect Root ($M = 0.99, SD = 0.15$) performs better than Expand and RaySelection, but worse than LenSelect Combined and IntenSelect. For the first time a difference between both LenSelect techniques can be observed. LenSelect Combined seems better suited to selection of far away objects.
A significant effect can be observed for the task completion time as well ($F(4, 486.29) = 93.75, p < 0.001, \eta^2 = 0.537$). Again Expand ($M = 1.94, SD = 0.98$) needs significantly more time for a selection than all other selection techniques. LenSelect Combined ($M = 0.55, SD = 0.26$) performs significantly better than all other selection techniques, except for IntenSelect ($M = 0.63, SD = 0.31$). RaySelection ($M = 0.62, SD = 0.21$), LenSelect Root ($M = 0.63, SD = 0.26$) and IntenSelect show no significant difference between each other. LenSelect Combined also leads to better selection times than LenSelect Root.

Rotating Cans:
With RaySelection ($n = 100$), LenSelect Root ($n = 100$), IntenSelect ($n = 95$), LenSelect Combined ($n = 100$) and Expand ($n = 83$).
The selection errors for the "Rotating Cans" tests scenario are significant as well. ($F(4, 229.0) = 3.01, p = 0.019, \eta^2 = 0.056$). Here IntenSelect ($M = 0.85, SD = 1.56$) shows significantly more selection errors than all other selection techniques. While RaySelection ($M = 0.32, SD = 0.79$), LenSelect Root ($M = 0.27, SD = 0.63$), LenSelect Combined ($M = 0.3, SD = 0.54$) and Expand ($M = 0.27, SD = 0.77$) show no significant effect compared to each other.
The task completion time for this test scenario shows a significant effect ($F(4, 223.26) = 28.54, p < 0.001, \eta^2 = 0.190$). Expand ($M = 2.36, SD = 1.07$) shows significantly higher selection times than the other selection techniques. IntenSelect ($M = 1.76, SD = 1.75$) shows the second longest selection times after Expand and takes significantly longer than RaySelection ($M = 1.16, SD = 0.56$), LenSelect Root ($M = 1.14, SD = 0.59$ and LenSelect Combined ($M = 1.09, SD = 0.42$). The latter don't show significant differences between them. IntenSelect's higher selection times can clearly be attributed to it's significant amount of selection errors for this test scenario. A similar effect can be observed in the "Cluttered Cans" test scenario, but for the sake of brevity it's not shown here.

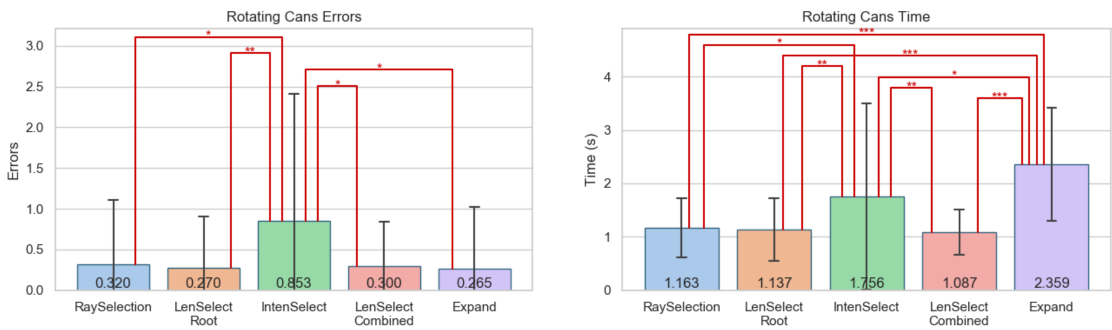


Figure 57.: Vive Study: Average selection errors and task completion time for the "Rotating Cans" test scenario per selection technique.

In addition to this data the practice times were stopped for each selection technique. As can be seen IntenSelect has a higher average this correlates with the conclusions from the questionnaire. But does not have to mean much. If the selection technique was more fascinating than the others, participants might have wanted to spent more time with it. RaySelection, naturally, shows the lowest practice times, this fits well with it's high rating on intuitive-use. While Expand shows the longest practice time, which is logical, since it is the most complex selection technique. LenSelect Root and Combined are the closest to RaySelection, when it comes to practice times.

Selection Technique	Mean	Median	Standard Deviation
RaySelection	14.01 s	14.79 s	8.71 s
LenSelect Root	17.52 s	14.39 s	12.12 s
IntenSelect	32.06 s	24.84 s	27.72 s
LenSelect Combined	18.50 s	12.48 s	20.46 s
Expand	48.46 s	19.70 s	58.00 s

Table 13.: Vive Study: Average time spent in practice mode for each selection technique.

Participant's Comments:
Some participants noted that overlapping objects got too big and made selection more difficult for LenSelect. It also made it hard to distinguish the cubes in the "Stacked Cubes" test scenario. One participant found it useful when selecting small objects.
With IntenSelect participants noted it was not entirely clear which object is currently hovered. One participant was frustrated since pointing at a can with the white ray not necessarily means it could be selected with IntenSelect. The cans were incredibly difficult to select for participants. One participant found the red ray a bit menacing.
For Expand participants remarked that it was hard to find the correct object in the grid if objects looked similar. Another noted using this technique is exhausting. A different participant exclaimed it actively obstructed them in reaching their goals and was aggravating to use. The question why all objects are in the grid, even though not all of them can be selected came up multiple times. The objects inside the grid were also small and cumbersome to select and their placement seemed arbitrary.
One participant found it incredibly difficult to select the cigarette package in the "Miscellaneous" test scenarios with Expand. It took them 68 seconds and they made 24 selection errors. This sample was removed from the study due to being such a vast outlier. But it still shows how flawed Expand is as a selection technique.
Also reducing the opacity of the lens seemed to be a success as one participant exclaimed they were not even of it, but it's still clearly visible.

Results:
Both LenSelect techniques are popular with users and seem as intuitive as a simple ray cast. They outperform both IntenSelect and Expand in terms of intuitive use. Seeming more familiar to participants while also needing less time to learn and less thought to use. The user ranking shows them to be more popular than Expand and a slight tendency of being more popular than IntenSelect and RaySelection.
On average they produce a slightly higher ID than IntenSelect (~ 0.2). However LenSelect Combined produces slightly better selection times ($\sim 70ms$), while LenSelect Root shows similar selection times to IntenSelect. They also show about the

same amount of errors as the other selection techniques, except for RaySelection, which is outperformed by all of them. Taking a closer look reveals that LenSelect is better suited to certain selections, while IntenSelect is better suited to others. Even the scaling functions show a better performance for some scenarios between each other, but most of the time they perform similarly.
It must also be noted that, for the sake of brevity, not all results for the test scenarios are shown. Only those indicative of a general trend. This means IntenSelect usually shows a lower ID for all test scenarios, with a few exceptions. While at the same time usually failing to produce better selection times. Usually IntenSelect and LenSelect perform equally well. The same goes for selection errors.

6.5.2. Powerwall

The study had n = 17 participants, 15 of which self-identified as male and two as female. Most of them were either research or scientific assistants or students of Computer Science. One person was left handed and none had a form of color blindness. Their ages ranged from 20 to 33 with a mean age of 25.42, a standard deviation of 3.62 and a median age of 24.5.
Participants were asked to rate the following statements on a Likert Scale ranging from 1 to 5, where 5 is complete agreement and 1 complete disagreement:

Question	Mean	Median	Standard Deviation
I'm playing video games often.	3.59	4	1.141
I'm often playing first person shooters.	2.71	2	1.125
I have experience with 3D-pointing devices.	3.94	4	1.110
I'm skilled at video games.	3.59	4	0.974
I have experience with HMD.	3.71	4	1.525

Table 14.: Powerwall Study: Familiarity of participants with the presented tasks.

Participants showed a high familiarity with the presented tasks.
Again due to COVID-19 finding participants was difficult. Therefore some participants from the Vive study were asked to participate again in the Powerwall study. Seven of the 15 participants in this study already participated in the previous study.

6.5.2.1. Questionnaire Data

One of the participants missed the question to rate the complexity for all selection techniques but Expand.

Complexity and Fun:

Complexity shows a significant effect ($H(4) = 32.02, p < 0.001$). Expand ($M = 3.41, SD = 0.94$) is rated significantly worse than all other selection techniques. RaySelection ($M = 1.50, SD = 0.52$) is rated significantly less complex than IntenSelect ($M = 2.63, SD = 0.81$). No significant difference was found between LenSelect Root ($M = 2.06, SD = 0.68$), LenSelect Combined ($M = 2.25, SD = 0.93$) and IntenSelect. Yet still, LenSelect is much closer to RaySelection than IntenSelect.
No significant effect was found for the fun rating ($H(4) = 8.76, p = 0.067$). In fact all results are almost identical, with the exception of Expand, which is rated a bit lower than the rest.

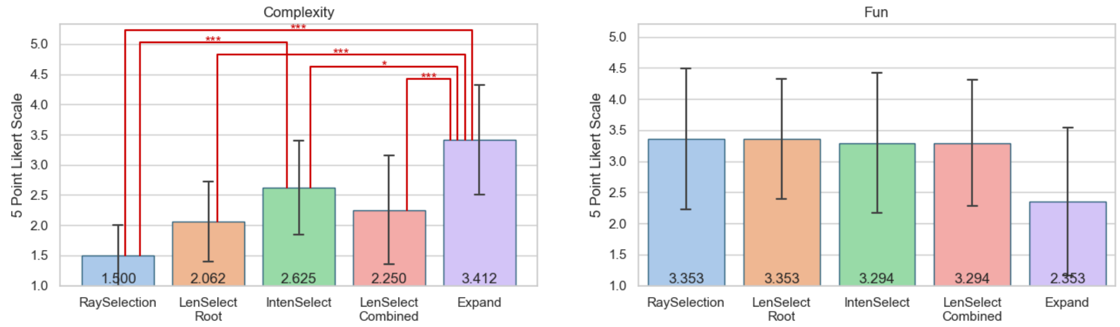


Figure 58.: Powerwall Study: Average complexity and fun per selection technique. A lower value is better for the complexity rating, while a higher value is better for the fun rating.

QUESI Total and User Ranking:

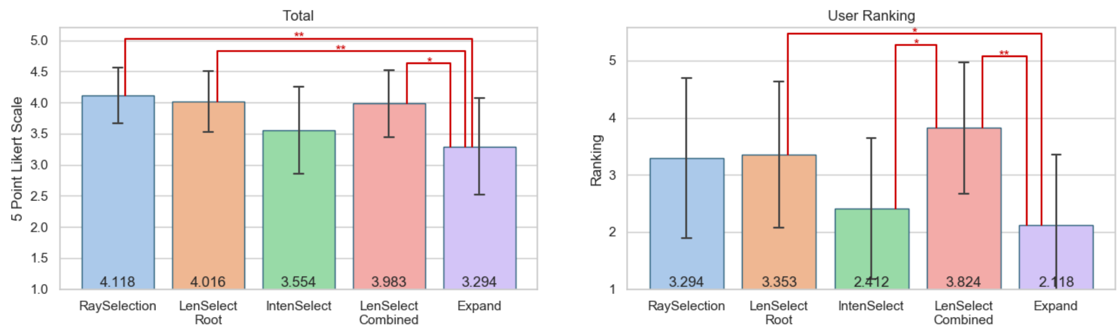


Figure 59.: Powerwall Study: QUESI Total and user ranking per selection technique. A higher value is better.

The Kruskal-Wallis test shows a significant effect ($H(4) = 16.41, p = 0.003$) for the QUESI Total. Expand ($M = 3.29, SD = 0.80$) is rated significantly worse than RaySelection ($M = 4.12, SD = 0.46$), LenSelect Root ($M = 4.02, SD = 0.50$) and LenSelect Combined ($M = 3.98, SD = 0.56$). Only IntenSelect ($M = 3.55, SD = 0.72$) shows no significant difference. In this study LenSelect is even closer to RaySelection

than the one before.
The user ranking showed a significant effect ($H(4) = 16.92, p = 0.002$). LenSelect Combined ($M = 3.82, SD = 1.19$) was rated significantly higher than IntenSelect ($M = 2.12, SD = 1.27$) and Expand ($M = 2.12, SD = 1.27$). LenSelect Root ($M = 3.35, SD = 1.32$) was rated significantly higher than Expand. RaySelection ($M = 3.29, SD = 1.45$) showed no significant difference between any selection technique and is rated about as high as LenSelect Root. LenSelect Combined is the most popular selection technique, followed closely by LenSelect Root and RaySelection.

Cognitive Load and Familiarity:

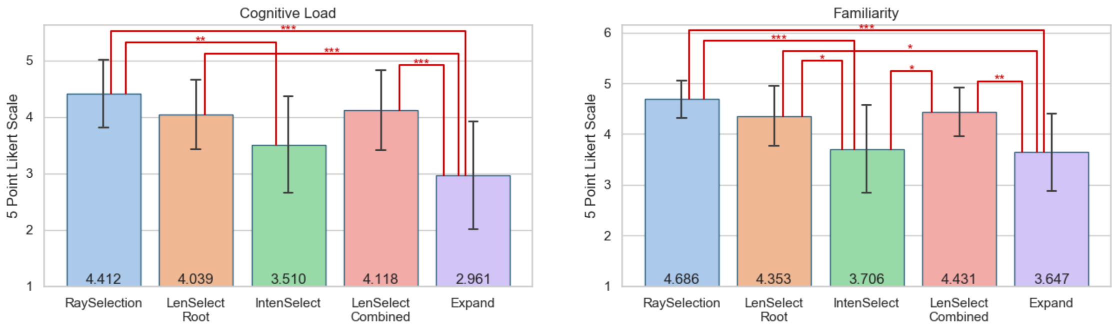


Figure 60.: Powerwall Study: Cognitive Load and Familiarity per selection technique. A higher value is better.

A significant effect ($H(4) = 23.45, p < 0.001$) could be observed for the Cognitive Load. Where RaySelection ($M = 4.41, SD = 0.62$) was rated significantly higher than IntenSelect ($M = 3.51, SD = 0.88$) and Expand ($M = 2.96, SD = 0.99$). There was also a significant difference between LenSelect Root ($M = 4.04, SD = 0.63$) and LenSelect Combined ($M = 4.12, SD = 0.73$) compared to Expand. IntenSelect ($M = 3.51, SD = 0.88$) was rated significantly worse than RaySelection. Ray Selection performs best in this scale, followed by LenSelect.
Familiarity also shows a significant effect ($H(4) = 23.10, p < 0.001$). Here RaySelection ($M = 4.69, SD = 0.38$) performs significantly better than IntenSelect ($M = 3.71, SD = 0.90$) and Expand ($M = 3.65, SD = 0.79$). LenSelect Root ($M = 4.35, SD = 0.61$) is rated significantly higher than Expand and IntenSelect. While LenSelect Combined ($M = 4.43, SD = 0.50$) is rated significantly higher than IntenSelect and Expand, as well.

Perceived Learning Effort:
A significant effect ($H(4) = 29.00, p < 0.001$) was observed for the Perceived Learning Effect as well. RaySelection ($M = 4.86, SD = 0.29$) is rated significantly higher than IntenSelect ($M = 4.04, SD = 0.69$) and Expand ($M = 3.51, SD = 0.97$). LenSelect Root ($M = 4.49, SD = 0.55$) and LenSelect Combined ($M = 4.55, SD = 0.53$) also performed significantly better than Expand. No significant effect could be found

between LenSelect and IntenSelect. An almost non-existent dispersion can be seen for RaySelection here. Other techniques show a much higher dispersion, with only LenSelect Combied getting close to RaySelection.

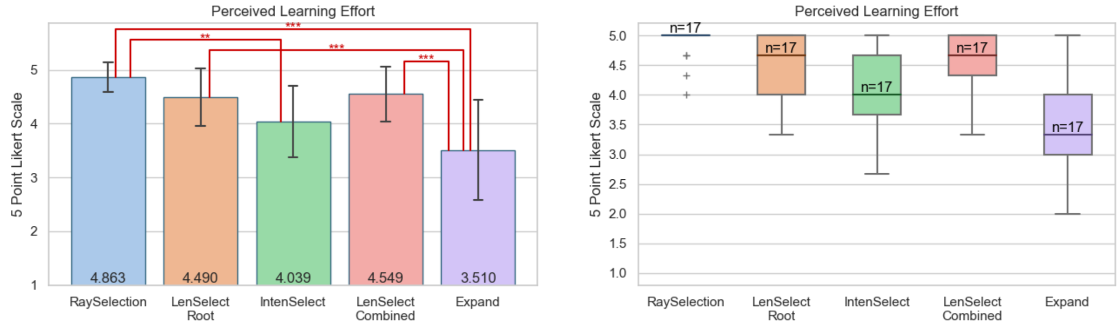


Figure 61.: Powerwall Study: Perceived Learning Effort per selection technique. A higher value is better.

Expand scores lowest of all selection techniques for the Powerwall as well. While RaySelection performs best when it comes to intuitive use, differences between it and both LenSelect techniques are never significant. While IntenSelect performs significantly worse in some metrics compared to RaySelection. Only rarely is the difference between IntenSelect and LenSelect significant.
LenSelect Combined is clearly preferred over IntenSelect and Expand, however and LenSelect Root is preferred over just Expand. It is unclear if both or one of them is preferred over RaySelection.

6.5.2.2. Fitts' Data

For this final study only half the cameras were available, this lead to worse tracking and some problems during selection.
13 samples were removed from this study due to distractions, questions and tracking issues. Two were removed for IntenSelect, two for RaySelection, three for LenSelect Combined, five for LenSelect Root and one for Expand.

This time IntenSelect shows the lowest intercept of selection techniques, but also a higher slope. LenSelect Combined has the lowest slope, while LenSelect Root has a slightly higher slope than RaySelection. LenSelect's intercept is also closer to that of RaySelection this time. This explains the results to an extent, with lower ID's selection times are very similar between selection techniques. Only when the ID gets higher the differences can be seen and felt.

	RaySelection	LenSelect Root	IntenSelect	LenSelect LenSelect Combined	Expand
a	0.406	0.416	0.232	0.542	0.906
b	0.306	0.353	0.561	0.244	0.494
TP_b	3.272	2.835	1.781	4.106	2.025

Table 15.: Vive Study: a, b and TP_b for different selection techniques.

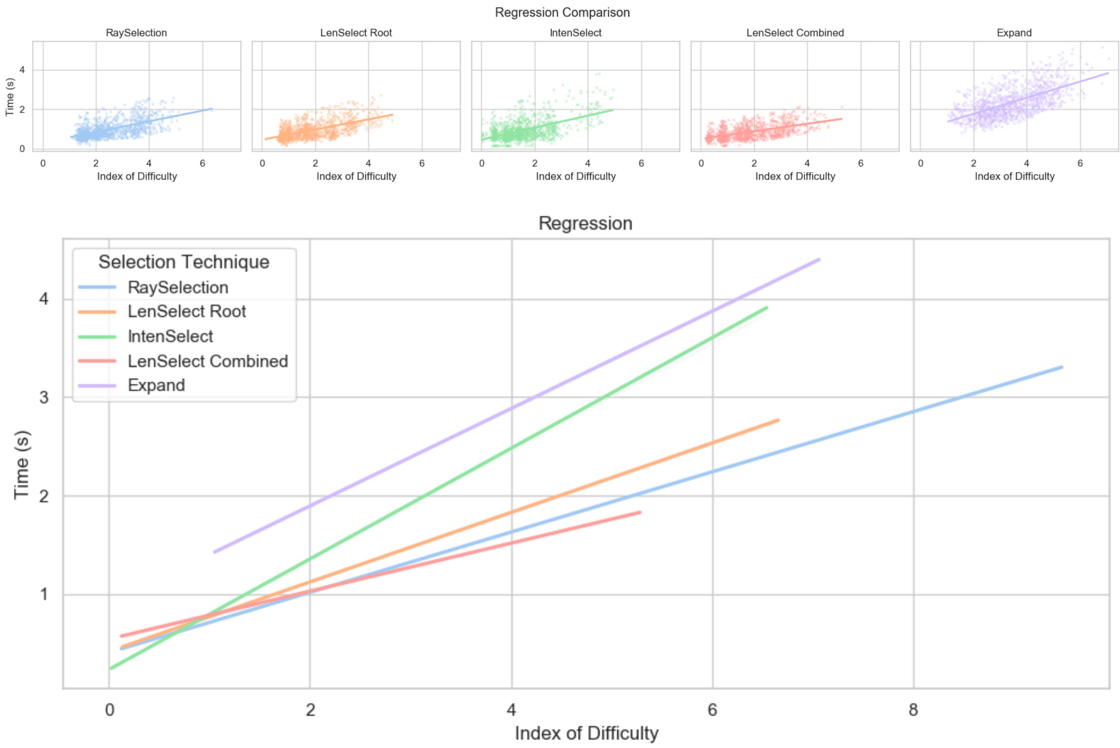


Figure 62.: Powerwall Study: Regression Data for all selection techniques.

Average Index of Difficulty and Task Completion Time:
The average Index of Difficulty shows a significant effect ($F(4, 3169.34) = 595.60, p < 0.001, \eta^2 = 0.297$) according to the Welch ANOVA. Every selection shows a significant difference compared to the others. Only LenSelect Root ($M = 1.88, SD = 1.01$) and LenSelect Combined ($M = 1.83, SD = 1.07$) show no such effect between each other. IntenSelect ($M = 1.60, SD = 1.03$) has the lowest ID, while Expand ($M = 3.48, SD = 1.17$) has the highest. RaySelection ($M = 2.51, SD = 0.97$) shows the second highest ID.
The task completion time also shows a significant effect ($F(4, 3120.03) = 334.23, p < 0.001, \eta^2 = 0.213$). Expand ($M = 2.63, SD = 1.46$) shows significantly higher selection times compared to all other selection techniques. While LenSelect Combined

$(M = 0.99, SD = 0.76)$ shows significantly lower selection times than all other techniques. RaySelection $(M = 1.17, SD = 1.01)$, LenSelect Root $(M = 1.08, SD = 0.94)$ and IntenSelect $(M = 1.13, SD = 1.56)$ show no significant differences between each other.

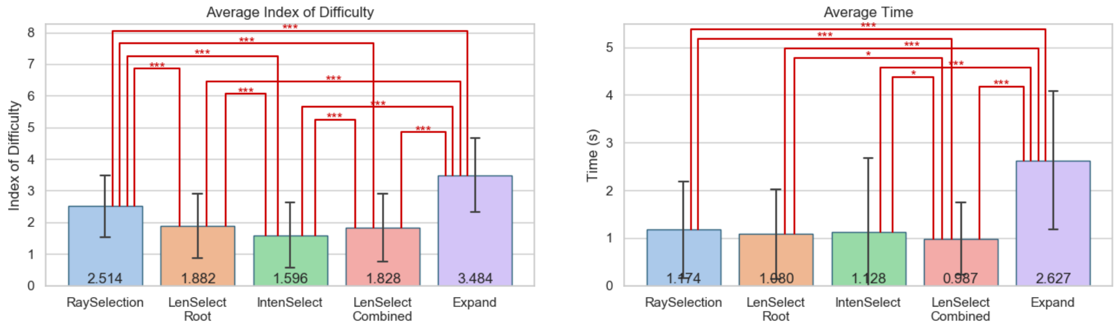
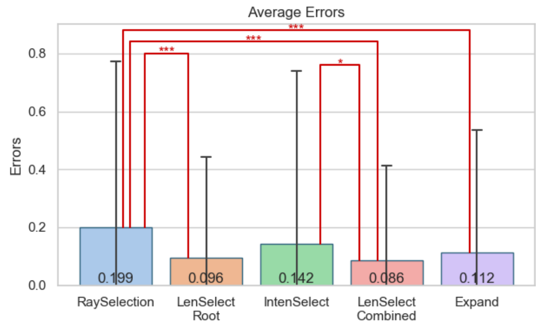


Figure 63.: Powerwall Study: Average Index of Difficulty and task completion time per selection technique.

Average Selection Errors:



A significant effect $(F(4, 3131.10) = 10.83, p < 0.001, \eta^2 = 0.008)$ was observed for the average selection errors, as well. Here RaySelection $(M = 0.20, SD = 0.57)$ shows significantly more errors compared to LenSelect Root $(M = 0.10, SD = 0.35)$, LenSelect Combined $(M = 0.09, SD = 0.33)$ and Expand $(M = 0.11, SD = 0.42)$. When removing the highest two outliers of nine and eight selection errors for IntenSelect $(M = 0.14, SD = 0.60)$ a significant difference between it and RaySelection could also be observed. LenSelect Combined shows significantly less errors than IntenSelect, however removing IntenSelect's outliers removes this effect as well.

Cluttered Cans:
With RaySelection $(n = 85,)$, LenSelect Root $(n = 83,)$, IntenSelect $(n = 85)$, LenSelect Combined $(n = 85)$ and Expand $(n = 79)$
Even though the "Cluttered Cans" test scenario shows no significant effect $(F(4, 199.44) = 1.99, p = 0.098, \eta^2 = 0.032)$ a slight tendency for higher errors can be observed for IntenSelect.

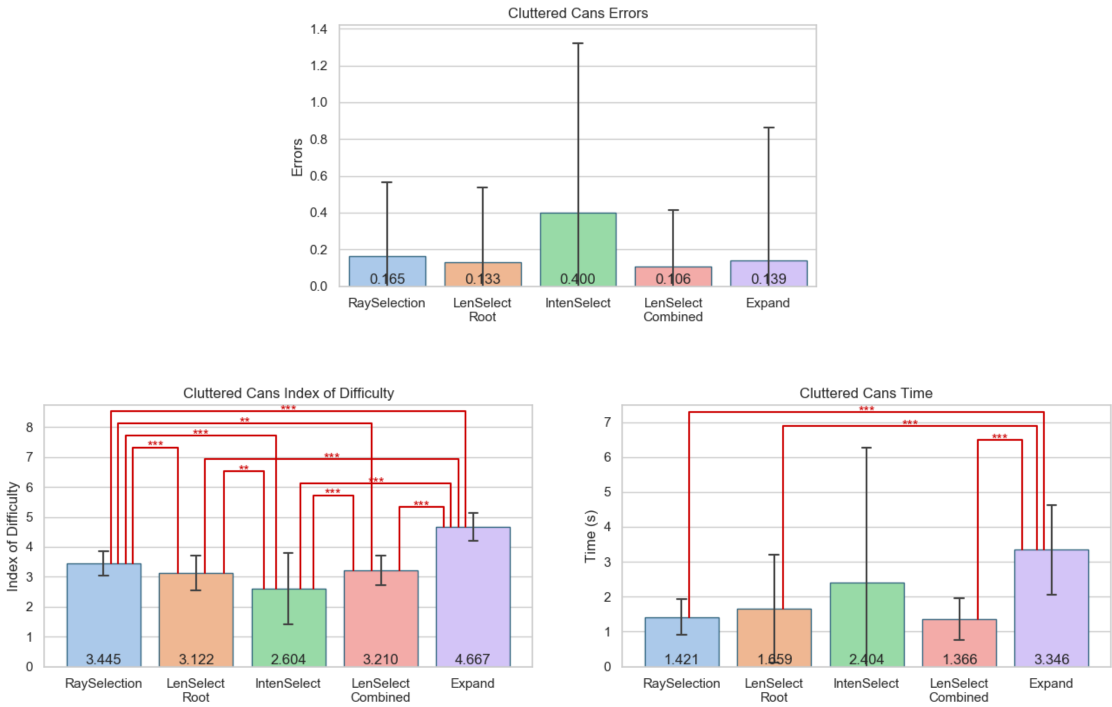


Figure 66.: Powerwall Study: Average Index of Difficulty and task completion time for the "Cluttered Cans" test scenario per selection technique.

For the Index of Difficulty a significant effect was found ($F(4, 202.93) = 151.87, p < 0.001, \eta^2 = 0.492$). Again all selection techniques show a significant difference compared to each other, only LenSelect Root ($M = 3.12, SD = 0.58$) and LenSelect Combined ($M = 3.21, SD = 0.49$) show no such effect between each other. Expand ($M = 4.67, SD = 0.46$) has the highest ID, while IntenSelect ($M = 2.60, SD = 1.20$) has the lowest. RaySelection ($M = 3.45, SD = 0.40$) performed better than Expand, but worse than the other selection techniques.
The task completion time also shows a significant effect ($F(4, 193.21) = 41.97, p < 0.001, \eta^2 = 0.121$). But here only Expand ($M = 3.35, SD = 1.30$) performs significantly worse than all other selection techniques, except when compared to IntenSelect ($M = 2.40, SD = 3.90$). Removing the outlier time-sample for IntenSelect also reveals a significant effect between it and Expand, as well as LenSelect Combined ($M = 1.37, SD = 0.60$) and RaySelection ($M = 1.42, SD = 0.52$), meaning it performs worse than those two. Otherwise groups show no significant differences, including LenSelect Root ($M = 1.66, SD = 1.55$). IntenSelect's higher selection times again seem to be a result of higher selection errors.

Erratic Spheres :
With RaySelection ($n = 85,$), LenSelect Root ($n = 85,$), IntenSelect ($n = 85$), LenSelect Combined ($n = 84$) and Expand ($n = 84$)
The ID for the "Erratic Spheres" test scenario shows a significant effect ($F(4, 207.91) =$

$211.29, p < 0.001, \eta^2 = 0.611$). Again there are significant differences between all selection techniques, except between LenSelect Root ($M = 2.19, SD = 0.64$) and LenSelect Combined ($M = 2.27, SD = 0.78$). Expand ($M = 4.07, SD = 0.52$) has the highest ID. While IntenSelect ($M = 1.60, SD = 0.72$) has the lowest. RaySelection ($M = 2.98, SD = 0.70$) performs better than Expand, but worse than both LenSelect techniques.
A significant effect ($F(4, 203.02) = 17.78p < 0.001, \eta^2 = 0.213$) was found for the task completion time of the "Erratic Spheres" test scenario. But the only significant difference was found between Expand ($M = 2.94, SD = 1.83$) and all other selection techniques. No other effect was found for RaySelection ($M = 1.37, SD = 1.50$), LenSelect Root ($M = 1.15, SD = 0.74$), IntenSelect ($M = 1.20, SD = 0.79$) and LenSelect Combined ($M = 1.30, SD = 1.32$). This is peculiar, as the Index of Difficulty showed so many significant differences between selection techniques and selection errors ($F(4, 206.05) = 1.73, p = 0.144, \eta^2 = 0.017$) showed no significant effect for this test scenario.

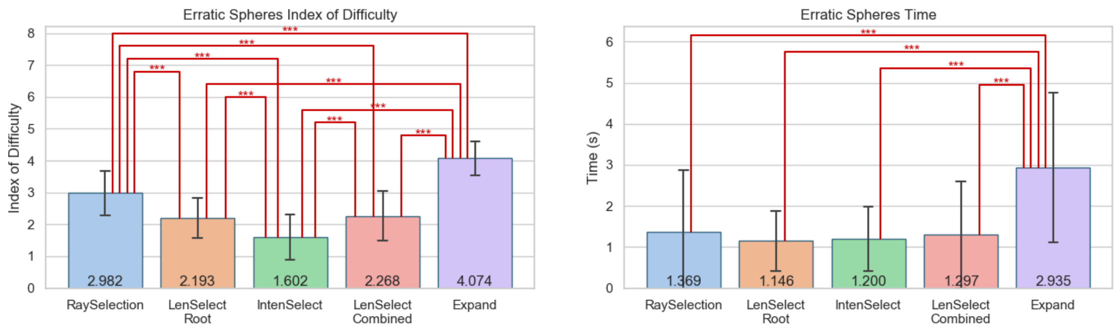


Figure 67.: Powerwall Study: Average Index of Difficulty and task completion time for the "Erratic Spheres" test scenario per selection technique.

Fast Sphere:
With $n = 85$.
For this test scenario IntenSelect performs the best
A significant effect ($F(4, 204.12) = 8.92, p < 0.001, \eta^2 = 0.106$) was found for the selection errors scale of the "Fast Sphere" test scenario. Here RaySelection ($M = 0.74, SD = 1.24$) shows significantly more selection errors than LenSelect Root ($M = 0.28, SD = 0.63$), IntenSelect ($M = 0.06, SD = 0.32$), LenSelect Combined ($M = 0.28, SD = 0.50$) and Expand ($M = 0.13, SD = 0.37$). While IntenSelect shows significantly less selection errors than LenSelect Root and LenSelect Linear. IntenSelect shows the least amount of errors , followed by Expand.
Another significant effect ($F(4, 206.02) = 376.36, p < 0.001, \eta^2 = 0.696$ could be found for the Index of Difficulty as well. Here Expand ($M = 3.68, SD = 0.38$) shows a significantly higher ID than all other selection techniques. RaySelection ($M = 3.05, SD = 0.79$) shows a significantly higher ID than the remaining selection techniques. And IntenSelect ($M = 0.90, SD = 0.56$) show a significantly lower

ID than LenSelect Root ($M = 2.29, SD = 0.56$) and LenSelect Combined ($M = 2.29, SD = 0.64$). IntenSelect shows by far the lowest ID, while both LenSelect techniques don't shows a significant difference between each other.
A significant effect ($F(4, 207.21) = 41.66, p < 0.001, \eta^2 = 0.20$) could also be found for the selection times of the "Fast Sphere" test scenario. Here Expand ($M = 2.31, SD = 0.80$) performs significantly worse than all other selection techniques, except for RaySelection ($M = 2.01, SD = 1.79$). While RaySelection shows much higher selection times than LenSelect Root ($M = 1.23, SD = 1.01$), IntenSelect ($M = 0.94, SD = 0.88$) and LenSelect Combined ($M = 1.04, SD = 0.70$). While those techniques did not show a significant difference between each other. Again IntenSelect seems to be unable to produce better selection times, even though it's ID was much better than those of the others.

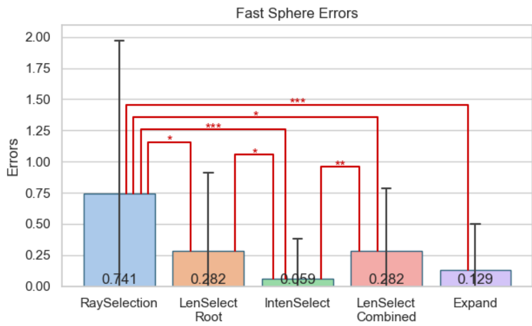


Figure 68.: Powerwall Study: Average selection errors for the "Fast Sphere" test scenario per selection technique.

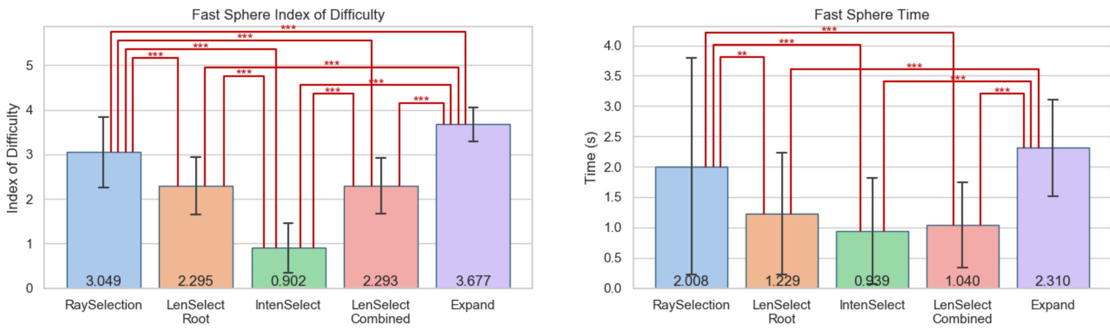


Figure 69.: Powerwall Study: Average selection errors and task completion time for the "Fast Sphere" test scenario per selection technique.

Propane Tanks Close:
Where Expand ($n = 168$), the rest $n = 170$.
For the "Propane Tanks Close" test scenario LenSelect Root shows the lowest ID, but also fails to produce lower selection times.
The Index of Difficulty shows a significant effect ($F(4, 409.21) = 415.15, p < 0.001, \eta^2 =$

0.895). Here Expand ($M = 2.53, SD = 0.41$) produces a significantly higher ID than all other selection techniques. RaySelection ($M = 1.48, SD = 0.16$) produces a significantly higher ID than the remaining selection techniques. While IntenSelect ($M = 0.97, SD = 0.12$) performs significantly worse than LenSelect Root ($M = 0.76, SD = 0.14$) and LenSelect Combined ($M = 0.87, SD = 0.14$).
But both fail to produce lower selection times. The task completion time is also significant ($F(4, 409.14) = 102.28, p < 0.001, \eta^2 = 0.353$). Only Expand ($M = 1.78, SD = 0.68$) shows a significant difference. It performs worse than RaySelection ($M = 0.78, SD = 0.94$), LenSelect Root ($M = 0.66, SD = 0.29$), IntenSelect ($M = 0.66, SD = 0.41$) and LenSelect Combined ($M = 0.68, SD = 0.38$). So LenSelect can fail in producing lower selection times when showing smaller ID's as well.

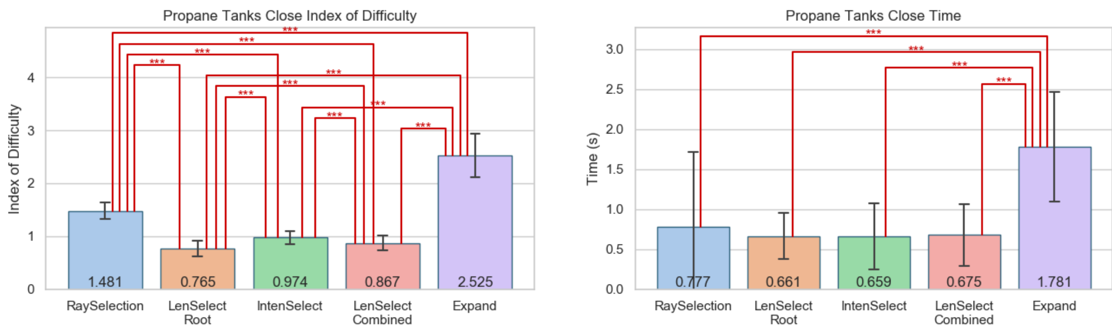


Figure 70.: Powerwall Study: Average Index of Difficulty and task completion time for the "Propane Tanks Close" test scenario per selection technique.

Propane Tanks Far:

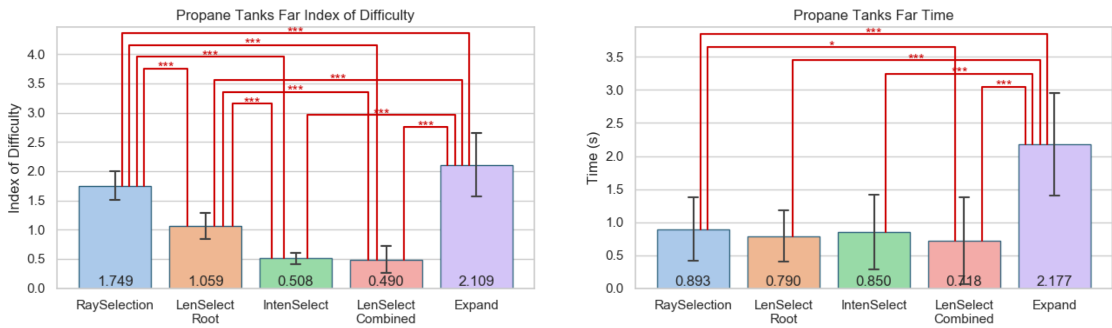


Figure 71.: Powerwall Study: Average Index of Difficulty and task completion time for the "Propane Tanks Far" test scenario per selection technique.

Where RaySelection, LenSelect Combined and Expand ($n = 170$) and LenSelect Root and IntenSelect ($n = 196$).
The Index of Difficulty shows a significant effect ($F(4, 392.58) = 1339.70, p < 0.001, \eta^2 = 0.819$). Expand ($M = 2.11, SD = 0.55$) performs worse than all

other selection techniques again. RaySelection ($M = 1.75, SD = 0.25$) shows significantly higher ID's than the remaining selection techniques. While LenSelect Root ($M = 1.06, SD = 0.22$) produces significantly higher ID's than IntenSelect ($M = 0.51, SD = 0.10$) and LenSelect Combined ($M = 0.49, SD = 0.23$). Both IntenSelect and LenSelet Combined provide the lowest ID.
But both fail to produce lower selection times than LenSelect Root.
For the task completion time a significant effect could be observed ($F(4, 415.82) = 120.55, p < 0.001, \eta^2 = 0.47$). Here Expand ($M = 2.18, SD = 0.77$) shows significantly higher selection times than all other selection techniques. While RaySelection ($M = 0.89, SD = 0.48$) shows significantly higher selection times than LenSelect Combined ($M = 0.72, SD = 0.77$). IntenSelect ($M = 0.85, SD = 0.57$) and LenSelect Root ($M = 0.79, SD = 0.39$) perform significantly better than Expand. This time both IntenSelect and LenSelect Combined failed to produce lower selection times, despite their lower ID.

Selection Technique	Mean	Median	Standard Deviation
RaySelection	12.90 s	5.44 s	13.75 s
LenSelect Root	16.70 s	14.23 s	13.62 s
IntenSelect	38.26 s	18.89 s	60.76 s
LenSelect Combined	22.72 s	15.44 s	24.66 s
Expand	28.94 s	16.01 s	50.88 s

Table 16.: Powerwall Study: Average time spent in practice mode for each selection technique.

The practice mode times basically show the same result as for the Vive study and the same conclusions apply.

Results:
The Powerwall study yielded similar results as the Vive study. LenSelect performs similar to IntenSelect and depending on the scenario one is better then the other. A lot of the same results can be observed here for the same test scenarios. Even the time spend in practice mode is similar, only longer. This might be because it's harder to get accustomed to the Powerwall and the uncommon way to operate the selection techniques for it. Again not all significant figures are shown. Usually the ID is a bit lower for IntenSelect compared to LenSelect. Generally LenSelect has a similar task completion time as IntenSelect. They also show about the same amount of selection errors.

Participant's Comments:
One participant noted that smaller objects could be scaled a bit bigger, while bigger objects could be scaled a bit smaller for their taste, for LenSelect Root.

It was noted that IntenSelect's red ray penetrating other objects makes it hard to see the currently hovered object. The cursor jumps too much between small objects, noted another participant.
For LenSelect Combined, object's sometimes became too big. The cans were easy to select with this technique.
While searching the object in Expand's grid was exhausting. The technique feels noticeably slower due to the additional click needed. Another participant found it useful for moving objects.
One participant wished to have RaySelection as the default and being able to toggle to LenSelect when necessary.

Conclusion:
Although IntenSelect performed among the best according to Fitts' Data, it was not the most accepted selection technique. Both LenSelect techniques clearly outperform it in almost all QUESI scales and are more popular than it. Meanwhile RaySelection is the most intuitive selection technique. So Hypothesis 1 did not come true.
Hypothesis 2 did come true. Expand produces on average the longest task completion times. This applies to both Vive and Powerwall.
Overall LenSelect performs among the best of the tested selection techniques.

7. Discussion

Both LenSelect and IntenSelect often are among the best selection techniques in this study. Overall they perform somewhat similar, even though IntenSelect generally has a lower Index of Difficulty, it fails to produce lower selection times. LenSelect Root shows similar selection times, while LenSelect Combined outperforms IntenSelect, but only marginally with $\sim 40ms$ to $\sim 70ms$. Comparing their selection errors shows they, again, perform similarly. In the end it's hard to tell if any of these techniques is superior, as they all outperform each other in certain test scenarios. IntenSelect works best if selection is only mildly obstructed by other objects. Meanwhile LenSelect performs more consistently among test scenarios. LenSelect Root also performs very well with selection of far away objects.

But LenSelect was clearly rated more intuitive in it's use compared to IntenSelect and was more popular with participants. LenSelect often came close to intuitiveness to RaySelection, which is logically the most intuitive, due to it's simplicity. Expand on the other hand performed the worst on almost all scales, except for selection errors. Where it performed comparably to LenSelect and IntenSelect.

LenSelect performed equally well compared to IntenSelect, a state-of-the-art selection technique. And both outperformed RaySelection when it comes to selection errors, task completion time and Index of Difficulty.

It is unclear how exactly these scales apply in practice, however. Is it better to have a lower ID, even if this does not lower the task completion time? Is it better to have less selection errors but take longer for a selection. An overarching scale for comparison is missing, this makes comparison problematic.

7.1. Intuitive Use

So why does LenSelect seem more intuitive than IntenSelect? Even though their basic premise is very similar. IntenSelect also produces no visual disruption. On first glance IntenSelect even produces a better target area than LenSelect. LenSelect simply resizes the object. A can, for example, will always have a bigger height than width. Therefore horizontal cursor movement will never show as low an ID as that of a vertical movement. IntenSelect, on the other hand, produces a circle around the object and the ID stays the same, no matter the vector of approach. But this only applies when there is enough space between objects. In a cluttered environment it can even lead to a reduced target area, as can be seen in Figure 72 (a). Taking this into account it's harder to predict where exactly the user needs to point to select an object.

Another explanation might be that directly pointing at the object does not select the object anymore. Instead the user basically tries to get as close to the center point of the object as possible. This works fine if there is enough space between object's. But when center point's of objects are overlapped by other object's the target area becomes harder to understand. When the target object's center point is overlapped by another object, a part of the target area is overlapped by this same object. Meaning the target can be selected through the overlapping object. On the other hand if the target object overlaps the center point of another object, pointing directly at this part of the object will select the overlapped object. If both of these effects happen simultaneously the target area might lie completely outside the visible part of the target object. This effect could be observed with the "Rotating Cans" test scenario, for example. This way the target area is completely obscured and a user aiming at the visible part of an object might get frustrated, since it won't select the desired object.
Lastly depending on how objects overlap IntenSelect's target areas can take on strange forms, further leading to frustration (Figure 72 (b)). Here some of the spheres were behind the target sphere leading to a "hole" in the target area. And two more spheres to either side squish the target area together.

However, rescaling the objects can lead to visual discomfort, so LenSelect is not without it's own problems. And the visual discomfort from LenSelect's scaling was not measured.

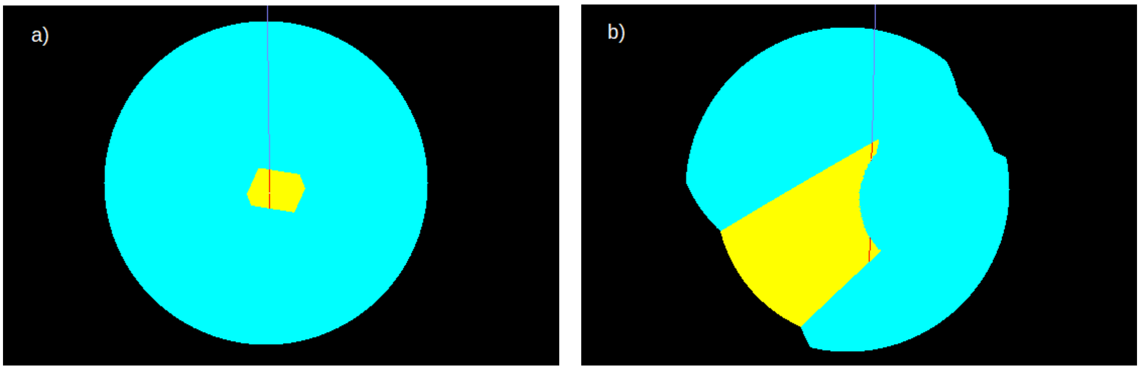


Figure 72.: IntenSelect's target areas a) "Rotating Cans" test scenario, b) "Erratic Spheres" test scenario

7.2. Caveats

Sometimes the mean also obfuscates the truth for the Index of Difficulty. Looking at a scatter plot for the "Cluttered Cans" test scenario (Figures 73 and 74) for example reveals distinct groups of selections.
This is because cans on the edges of the scenario have much bigger target areas than

those surrounded by other cans for IntenSelect. For LenSelect the cans in the front row are easier to select, since they are unobstructed by other objects.
What's interesting, is that we can see IntenSelect performing better for the group that's easier to select for it. But looking at the groups that are harder to select LenSelect performs better and more consistently. IntenSelect almost performs the worst for these selections, in fact.

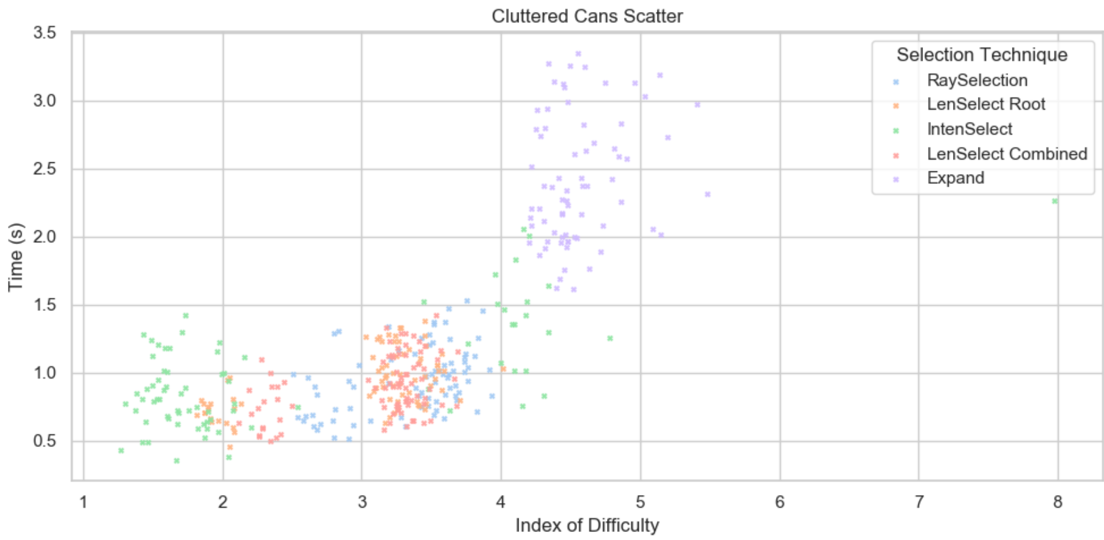


Figure 73.: Vive Study: Scatter plot for the "Cluttered Cans" test scenario. Revealing distinct groups of selections.

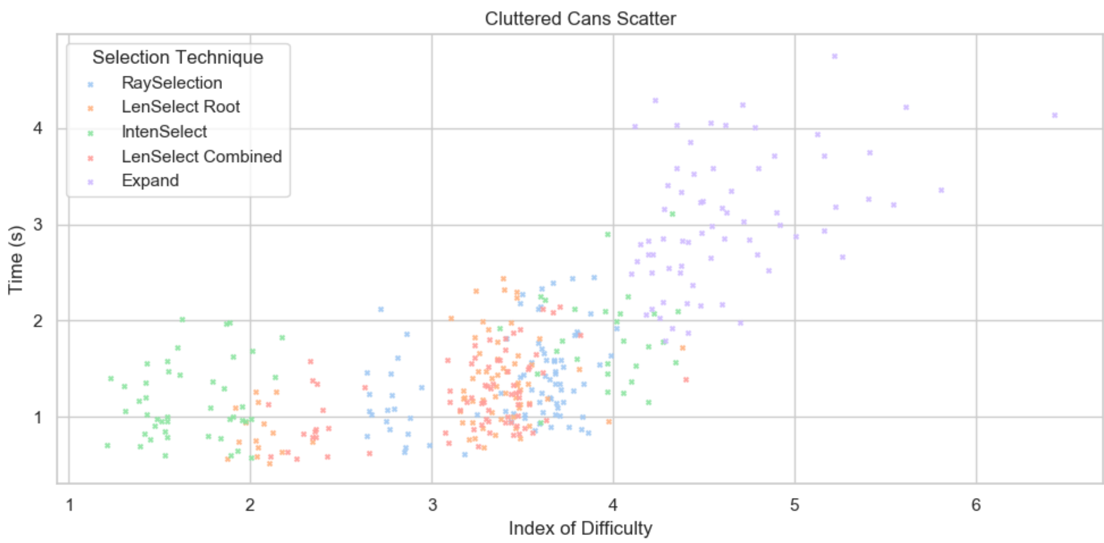


Figure 74.: Powerwall Study: Scatter plot for the "Cluttered Cans" test scenario. Revealing distinct groups of selections.

It's also interesting that a lower ID not always translates to a lower task completion time. This can have multiple causes. First if a selection error occurs the selection time will automatically be worse. This might be one explanation for test scenarios where IntenSelect produces many more mistakes than other selection techniques, like the "Rotating Cans" test scenario.
It is also of note that the Index of Difficulty is not always correct. For the "Stacked Cubes" test scenario the ID is calculated only after the pointer rests on the object, in the case of LenSelect. But the objects show big overlap during selection. In reality the pointer must get closer to the object than the ID in that scale shows. While IntenSelect has the same problem as with the "Cluttered Cans" test scenario where objects at the border deflate the overall ID for objects at the center of the test scenario.
Both the "Erratic Spheres" and "Fast Sphere" test scenarios are problematic as well. The former shows random overlaps, so the ID depends on the overall location of the spheres. It also shows strange target areas for IntenSelect, depending on the location of the spheres. Both these test scenarios can randomly and sometimes drastically change the selection task. Making it hard to find a definitive conclusion regarding task completion time and Index of Difficulty. But still, IntenSelect seems to perform better here than other selection techniques.

7.3. User Behaviour

Sometimes user's would move their hand above their head to get a better angle for selection. This happened with ray-based selection techniques and mostly for cluttered environments, such as the "Miscellaneous" or "Rotating Cans" test scenarios. Even though this only happened sporadically, not for every participant and not even consistently with the participant, it still shows a failure of the selection technique. It basically shows that users wish to move to get a better view and angle for the selection. Since they were not allowed to move away they instead moved their hand to produce a better angle.
This effect could also be observed more often with the Powerwall than the Vive, maybe the Powerwall had a flatter angle than the Vive.

7.4. Comparison Powerwall vs HTC Vive

Not much of a difference was found between Vive and Powerwall. All selection techniques perform slightly worse and sometimes results weren't as conclusive as those of the Vive study. But generally speaking the results are the same.

8. Future-Work

LenSelect seems generally promising and with further refinement of it's object scaling algorithm might even be able to outperform IntenSelect.
One such refinement could be dynamically changing the order of the root depending on the amount of objects inside the cone. The more objects the higher the order. This way selecting an object from a cluttered environment will become even easier.
A proper way to scale small objects bigger than big objects should also be researched. Currently the respective dimensions of the object divided by three is used for this, but doesn't really work well.
Removing objects from the scaling whose center point does not lie inside the lens might also be promising.
It should also be checked if the ID calculated from screen shots taken from the camera's and the controller's perspective is comparable.
Maybe a better way to calculate the ID could be worked out, as it doesn't really work for some of the test scenarios. The problem is that these types of algorithms depend heavily on the used selection technique. For example the current solution works fine for RaySelection.
Furthermore looking at Figure 48 on page 79, one could make the assumption that Fitts' Law does not apply to IntenSelect. As the scatter plot show higher dispersion the higher the ID becomes. But this is only speculation.

9. Conclusion

In this thesis LenSelect a new selection technique based on rescaling objects was evaluated. It shows a thorough examination of selection techniques. As stated in the beginning there is no superior selection technique, no selection technique can meet all requirements. But LenSelect, in either iteration, and IntenSelect proof to be the most promising of the tested selection techniques, even though both show quirks and annoyances. They all have their merits in certain circumstances, but also might need some further improvements. Especially LenSelect is still highly experimental and shows potential for further refinement.
It's unfortunate that due to COVID-19 the pool of study participants was reduced to, mainly, students and research assistants in the field of Computer Science and Digital Media. So participants had a high familiarity with VR and general understanding of controllers and their use in VR. In the end it's not possible to tell if LenSelect or IntenSelect is superior to the other.
However LenSelect showed to be more consistent among the presented test scenarios and was more popular with participants. It also shows a higher rating on intuitive-use, almost rivaling RaySelection.
With further refinement an easy to use, fast and reliable selection technique could be worked out.

10. Acknowledgements

I want to thank Marc Jochens for his support, especially in regards to the Unreal Engine and the Powerwall.
Furthermore I want to thank Joscha Cepok for the formula to find the closest vertex of a 3D-Object to a ray. As well as his feed back for the evaluation and figures used therein.
I also want to thank anyone who attended the group meetings for their valuable remarks and input.
I also want to thank all of the staff at the CGVR group of the University of Bremen for their support.

11. Eidesstattliche Erklärung

Hiermit erkläre ich, an Eides statt, dass die hier vorliegende Masterarbeit von mir selbst und ohne die (unzulässige) Hilfe Dritter angefertigt wurde und die benutzten Hilfsmittel vollständig angegeben sind. Ferner versichere ich, dass ich alle wörtlich und sinngemäß übernommenen Textstellen sowie fremde in der Arbeit enthaltene Zeichnungen, Skizzen und grafische Darstellungen gemäß gängigen wissenschaftlichen Zitierregeln als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ort, Datum	Waldemar Wegele

Bibliography

[1] Accot, J. and Zhai, S. "Refining Fitts' Law Models for Bivariate Pointing". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 193–200. ISBN: 1-58113-630-7. DOI: 10.1145/642611.642646. URL: http://doi.acm.org/10.1145/642611.642646.

[2] Araujo e Silva, F. B. de. "Increasing Selection Accuracy and Speed through Progressive Refinement". PhD thesis. Virginia Polytechnic Institute and State University, June 2015.

[3] Argelaguet, F. and Andujar, carlos. "A survey of 3D object selection techniques for virtual environments". In: *Computers and Graphics*. Elsevier, May 2013, pp. 121–136. DOI: https://doi.org/10.1016/j.cag.2012.12.00.

[4] Argelaguet, F., Andujar, C., and Trueba, R. "Overcoming Eye-hand Visibility Mismatch in 3D Pointing Selection". In: *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*. VRST '08. Bordeaux, France: ACM, 2008, pp. 43–46. ISBN: 978-1-59593-951-7. DOI: 10.1145/1450579.1450588. URL: http://doi.acm.org/10.1145/1450579.1450588.

[5] Bacim, F., Kopper, R., and Bowman, D. A. "Design and Evaluation of 3D Selection Techniques based on Progressive Refinement". In: *International Journal of Human-Computer Studies* 71(7-8) (July 2013), pp. 785–802.

[6] Baloup, M., Pietrzak, T., and Casiez, G. "RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300331.

[7] Benko, H. and Feiner, S. "Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection". In: *2007 IEEE Symposium on 3D User Interfaces*. Mar. 2007, pp. 79–86. DOI: 10.1109/3DUI.2007.340778.

[8] Bowman, D. A. "Interaction Techniques for Immersive Virtual Environments". In: *Design, Evaluation, and Application, Human-Computer Interaction Consortium Conference '98 (HCIC)* (1998), pp. 37–53. URL: https://www.researchgate.net/profile/Doug_Bowman/publication/2641457_Interaction_Techniques_for_Immersive_Virtual_Environments_Design_Evaluation_and_Application/links/0912f510a8cebb11ea000000.pdf (visited on 02/01/2020).

[9] Bowman, D. and Hodges, L. F. "An Evaluation of Techniques for Grabbing and Manipulating RemoteObjects in Immersive Virtual Environments". In: *Proceedings of the 1997 Symposium on Interactive 3DGraphics*. ACM, 1997, pp. 35–38.

[10] Bowman, D. A., Johnson, D. B., and Hodges, L. F. "Testbed Evaluation of Virtual Environment Interaction Techniques". In: *Presence: Teleoperators and Virtual Environments* 10(1) (2001), pp. 75–95. DOI: 10.1162/105474601750182333. eprint: https://doi.org/10.1162/105474601750182333. URL: https://doi.org/10.1162/105474601750182333.

[11] Carpendale, M. S. T. and Montagnese, C. "A Framework for Unifying Presentation Space". In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. UIST '01. Orlando, Florida: Association for Computing Machinery, 2001, pp. 61–70. ISBN: 158113438X. DOI: 10.1145/502348.502358. URL: https://doi.org/10.1145/502348.502358.

[12] Cashion, J., Wingrave, C., and LaViola Jr., J. J. "Dense and Dynamic 3D Selection for Game-Based Virtual Environments". In: *IEEE Transactions on Visualization and Computer Graphics* 4(18) (Apr. 2012), pp. 634–642.

[13] Casiez, G., Roussel, N., and Vogel, D. "1 € Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 2527–2530. ISBN: 9781450310154. DOI: 10.1145/2207676.2208639. URL: https://doi.org/10.1145/2207676.2208639.

[14] Cribbie, R. A., Fiksenbaum, L., Keselman, H. J., and Wilcox, R. R. "Effect of non-normality on test statistics for one-way independent groups designs". In: *British Journal of Mathematical and Statistical Psychology* 65(1) (2012), pp. 56–73. DOI: 10.1111/j.2044-8317.2011.02014.x.

[15] Douglas, S., Kirkpatrick, A., and MacKenzie, I. "Testing Pointing Device Performance and User Assessment with the ISO 9241, Part 9 Standard." In: Jan. 1999, pp. 215–222. DOI: 10.1145/302979.303042.

[16] Drewes, H. *A Lecture On Fitt's Law*. 2013. URL: http://www.cip.ifi.lmu.de/~drewes/science/fitts/A%20Lecture%20on%20Fitts%20Law.pdf (visited on 08/11/2019).

[17] Drewes, H. "Only One Fitts' Law Formula Please!" In: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '10. Atlanta, Georgia, USA: ACM, 2010, pp. 2813–2822. ISBN: 978-1-60558-930-5. DOI: 10.1145/1753846.1753867. URL: http://doi.acm.org/10.1145/1753846.1753867.

[18] Forlines, C., Balakrishnan, R., Beardsley, P., Baar, J. van, and Raskar, R. "Zoom-and-Pick: Facilitating Visual Zooming and Precision Pointing with Interactive Handheld Projectors". In: *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. UIST '05. Seattle, WA, USA: Association for Computing Machinery, 2005, pp. 73–82. ISBN: 1595932712. DOI: 10.1145/1095034.1095046. URL: https://doi.org/10.1145/1095034.1095046.

[19] Frees, S., Kessler, G. D., and Kay, E. "PRISM interaction for enhancing control in immersive virtual environments". In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 14(1) (May 2007). DOI: 10.1145/1229855.122985.

[20] Grossman, T. and Balakrishnan, R. "Pointing at Trivariate Targets in 3D Environments". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 447–454. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985749. URL: http://doi.acm.org/10.1145/985692.985749.

[21] Grossman, T. and Balakrishnan, R. "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*. UIST '06. Montreux, Switzerland: ACM, 2006, pp. 3–12. ISBN: 1-59593-313-1. DOI: 10.1145/1166253.1166257. URL: http://doi.acm.org/10.1145/1166253.1166257.

[22] Haan, G., Koutek, M., and Post, F. "IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection." In: Jan. 2005, pp. 201–209. DOI: 10.2312/EGVE/IPT_EGVE2005/201-209.

[23] Hajri, A. A., Fels, S., Miller, G., and Ilich, M. "Moving Target Selection in 2D Graphical User Interfaces". In: *Human-Computer Interaction – INTERACT 2011*. Ed. by P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 141–161. ISBN: 978-3-642-23771-3.

[24] Hoffmann, E. R. "Capture of moving targets: a modification of Fitts' Law". In: *Ergonomics* 34(2) (May 1990), pp. 211–220.

[25] Hoffmann, E. R. "Which Version/Variation of Fitts' Law? A Critique of Information-Theory Models". In: *Journal of Motor Behavior* 45(3) (2013). PMID: 23581725, pp. 205–215. DOI: 10.1080/00222895.2013.778815. eprint: https://doi.org/10.1080/00222895.2013.778815. URL: https://doi.org/10.1080/00222895.2013.778815.

[26] Honjing, L., Yanju, L., and Brooks Gordon, P. "Outlier Impact and Accommodation Methods:Multiple Comparisons of Type I Error Rates". In: *Journal of Modern Applied Statistical Methods* 15(1) (May 2016), pp. 452–471. DOI: 10.22237/jmasm/1462076520.

[27] Honjing, L., Yanju, L., and Brooks Gordon, P. "Outlier Impact and Accommodation on Power". In: *Journal of Modern Applied Statistical Methods* 16(1) (May 2017), pp. 261–278. DOI: 10.22237/jmasm/1462076520.

[28] J, W., J, G., A, C., and Bechmann, D. "Starfish: a selection technique for densevirtual environments". In: *Proceedings of the ACM symposium on virtual reality software andtechnology*, pp. 101–104.

[29] Jagacinski, R. J., Repperger, D. W., Ward, S. L., and Moran, M. S. "A Test of Fitts' Law with Moving Targets". In: *Human Factors* 22(2) (1980), pp. 225–233. DOI: 10.1177/001872088002200211. eprint: https://doi.org/10.1177/001872088002200211. URL: https://doi.org/10.1177/001872088002200211.

[30] Jenny, B. and Kelso, N. V. "Color Design for the Color Vision Impaired". In: *Cartographic Perspectives* (58) (2007), pp. 61–67. DOI: https://doi.org/10.14714/CP58.270.

[31] Kopper, R., Bowman, D. A., Silva, M. G., and McMahan, R. P. "A human motor behavior model for distal pointing tasks". In: *International Journal of Human-Computer Studies*. Vol. 68. 10. Elsevier, Oct. 2010, pp. 603–615.

[32] Lacoche, J., Duval, T., Arnaldi, B., Maisel, E., and Royan, J. "Machine Learning Based Interaction Technique Selection for 3D User Interfaces". In: *Virtual Reality and Augmented Reality*. Ed. by P. Bourdot, V. Interrante, L. Nedel, N. Magnenat-Thalmann, and G. Zachmann. Cham: Springer International Publishing, 2019, pp. 33–51. ISBN: 978-3-030-31908-3.

[33] Lu, Y., Yu, C., and Shi, Y. "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)*. 2020, pp. 35–43. DOI: 10.1109/VR46266.2020.00021.

[34] MacKenzie, I. S. "Fitts' Law". In: *The Wiley Handbook of Human Computer Interaction*. Ed. by K. L. Norman and J. Kirakowski. Hoboken, NJ: Wiley, 2018. Chap. 17, pp. 349–370. ISBN: 9781118976135. DOI: 10.1002/9781118976005.

[35] MacKenzie, I. S. "Fitts' Law as a Research and Design Tool in Human-Computer Interaction". In: *HUMAN-COMPUTER INTERACTION* (1992), pp. 91–139.

[36] MacKenzie, I. S. and Buxton, W. "Extending Fitts' Law to Two-Dimensional Tasks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: ACM, 1992, pp. 219–226. ISBN: 0-89791-513-5. DOI: 10.1145/142750.142794.

[37] MacKenzie, I. S. "A Note on the Validity of the Shannon Formulation for Fitts' Index of Difficulty". In: *Open Journal of Applied Sciences* 3(6) (2013), pp. 360–368. DOI: 10.4236/ojapps.2013.36046.

[38] Mendes, D., Medeiros, D., Cordeiro, E., Sousa, M., Ferreira, A., and Jorge, J. "PRECIOUS! Out-of-reach selection using iterative refinement in VR". In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. Mar. 2017, pp. 237–238. DOI: 10.1109/3DUI.2017.7893359.

[39] Mine, M. R. *Virtual Environment Interaction Techniques*. Technical Report. Chapel Hill, NC: University of North Carolina, Department of Computer Science, Mar. 1995.

[40] Mircioiu, C. and Atkinson, J. "A Comparison of Parametric and Non-Parametric Methods Applied to a Likert Scale". In: *Pharmacy* 5(2) (May 2017). DOI: https://doi.org/10.3390/pharmacy5020026.

[41] Murata, A. and Iwase, H. "Extending Fitts' law to a three-dimensional pointing task". In: *Human Movement Science* (2001). DOI: https://doi.org/10.1016/S0167-9457(01)00058-6.

[42] Naumann, A. and Hurtienne, J. "Benchmarks for Intuitive Interaction with Mobile Devices". In: *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI '10. Lisbon, Portugal: ACM, 2010, pp. 401–402. ISBN: 978-1-60558-835-3. DOI: 10.1145/1851600.1851685. URL: http://doi.acm.org/10.1145/1851600.1851685.

[43] Ortega, M. "Hook: Heuristics for selecting 3D moving objects in dense target environments". In: *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. Mar. 2013, pp. 119–122. DOI: 10.1109/3DUI.2013.6550208.

[44] Poupyrev, I., Weghorst, S., Billinghorst, M., and Ichikawa, T. "A Framework and Testbed for Studying Manipulation Techniques for Immersive VR". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 1997, pp. 21–28.

[45] Rohde, F. A. "Design, Implementierung und Evaluation neuartiger 3D Selektionsmetaphern in Virtual Reality". German. Bachelor Thesis. University of Bremen, 2018.

[46] Sarkar, M. and Brown, M. H. "Graphical Fisheye Views of Graphs". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: Association for Computing Machinery, 1992, pp. 83–91. ISBN: 0897915135. DOI: 10.1145/142750.142763. URL: https://doi.org/10.1145/142750.142763.

[47] Shingala, M. C. and Rajyaguru, A. "Comparison of Post Hoc Tests for Unequal Variance". In: *International Journal of New Technologies in Science and Engineering* 2 (Nov. 2015), pp. 22–33. ISSN: 2349-0780.

[48] Shoemaker, G., Tsukitani, T., Kitamura, Y., and Booth, K. S. "Two-Part Models Capture the Impact of Gain on Pointing Performance". In: *ACM Transactions on Computer-Human Interactions* 19(4) (Dec. 2012), 28:1–28:34. ISSN: 1073-0516. DOI: 10.1145/2395131.2395135. URL: http://doi.acm.org/10.1145/2395131.2395135.

[49] *3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays*. 8th International Immersive Projection Technology Workshop. 2004, pp. 163–170.

[50] Tabachnik, B. G. and Fidell, L. S. *EXPERIMENTAL DESIGNS USING ANOVA*. Belomnt, CA: Duxbury Press, 2007.

[51] Teather, R. J. "Evaluating 3D Pointing Techniques". PhD thesis. Toronto, Ontario: York University, 2013.

[52] Tovi, G. and Balakrishnan, R. "The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area". In: *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Apr. 2005, pp. 281–290. DOI: https://doi.org/10.1145/1054972.1055012.

[53] Tovi, G. and Balakrishnan, R. "The design and evaluation of selection techniques for 3D volumetric displays". In: *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, Oct. 2006, pp. 3–12. DOI: https://doi.org/10.1145/1166253.1166257.

[54] Ulrich, D. "Intuitive Interaktion: Eine Exploration von Komponenten, Einflussfaktoren und Gestaltungsansätzen aus der Perspektive des Nutzererlebens". German. Dissertation. Technische Universität Darmstadt, May 2014, pp. 1–180.

[55] Vallat, R. *Pingouin: Guidelines*. 2020. URL: https://pingouin-stats.org/guidelines.html.

[56] Vallat, R. "Pingouin: statistics in Python". In: *The Journal of Open Source Software* 3(31) (Nov. 2018), p. 1026.

[57] Vanacken, L., Grossman, T., and Coninx, K. "Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments". In: *2007 IEEE Symposium on 3D User Interfaces*. 2007, pp. 117–124. DOI: 10.1109/3DUI.2007.340783.

[58] Vanacken, L., Grossman, T., and Coninx, K. "Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments". In: *2007 IEEE Symposium on 3D User Interfaces*. Mar. 2007, pp. 115–122. DOI: 10.1109/3DUI.2007.340783.

[59] Welford, A. T., Norris, A. H., and Shock, N. W. "Speed and accuracy of movement and their changes with age". In: *Acta Psychologica* 30 (1969), pp. 2–15.

[60] Wikipedia contributors. "Latin square". In: *Wikipedia, The Free Encyclopedia* (2020). URL: `https://en.wikipedia.org/w/index.php?title=Latin_square&oldid=943888076` (visited on 03/22/2020).

[61] Wolf, D., Gugenheimer, J., Combosch, M., and Rukzio, E. "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, 2020, pp. 1–10. ISBN: 9781450367080. DOI: `10.1145/3313831.3376876`. URL: `https://doi.org/10.1145/3313831.3376876`.

[62] Zhai, S. "Characterizing computer input with Fitts' law parameters — The information and non-information aspects of pointing". In: vol. 61. 6. Elsevier, Dec. 2004, pp. 791–809.

A. 3D-Object sources

Street cones
Adrian3dartist: http://www.sharecg.com/v/54478/browse/5/ 3D-Model/Street-cones

Oil barrel
Animated Heaven: PBR. http://www.sharecg.com/v/87275/browse/5/3D-Model/Oil-Barrel-PBR

City trash and waste set
Crumpler: https://www.unrealengine.com/zh-CN/ marketplace/city-trash-and-waste-set

Watering can
Dorland: http://www.sharecg.com/v/33034/browse/5/3D-Model/ Watering-can

B. Questionnaire

The questionnaire used for the study, reduced to one selection technique.

Selection Analysis - Questionnaire

Subject Number:________________________ Date: _______________

Gender: ________________________ Age: _______________

Dominant hand: ________________________ Color Blindness: _______________

Occupation: ________________________

Prequestionnaire

Rate the following statements:	Fully disagree	Mainly disagree	Neutral	Mainly Agree	Fully Agree
I'm playing videogames often.	O	O	O	O	O
I'm often playing first-person-shooters.	O	O	O	O	O
I have experience with 3D-pointing devices (Wii-Mote, PS-Move, Kinect, ...).	O	O	O	O	O
I'm skilled at videogames.	O	O	O	O	O
I'm comfortable pointing at objects with a controller.	O	O	O	O	O
I have experience with Head-Mounted-Displays (HTC Vive, Occulus Rift, PS VR, ...).	O	O	O	O	O
I can use my dominant hand without problem.	O	O	O	O	O

Selection Technique		1

How complex was the selection technique in your opnion?

Not at all	Barely	Moderate	Considerable	Exceedingly
O	O	O	O	O

Rate the following statements:	Fully disagree	Mainly disagree	Neutral	Mainly Agree	Fully Agree
I could use the selection technique without thinking about it.	O	O	O	O	O
I achieved what I wanted to achieve with the selection technique.	O	O	O	O	O
The way the selection technique worked was immediatly clear to me.	O	O	O	O	O
I could interact with the selection technique in a way that seemed familiar to me.	O	O	O	O	O
No problems occured when using the selection technique.	O	O	O	O	O
Using the selection technique was inspiring.	O	O	O	O	O
The selection technique was not complicated to use.	O	O	O	O	O
I was able to achieve my goals in the way I had imagined to.	O	O	O	O	O
The selection technique was easy to use from the start.	O	O	O	O	O
It was always clear to me what I had to do to use the selection technique.	O	O	O	O	O
The process of using the selection technique went smoothly.	O	O	O	O	O

Rate the following statements:	Fully disagree	Mainly disagree	Neutral	Mainly Agree	Fully Agree
Using the selection technique carried me away.	O	O	O	O	O
I barely had to concentrate on using the selection technique.	O	O	O	O	O
The selection technique helped to completely achieve my goals.	O	O	O	O	O
How the selection technqiue is used was was clear to me straight away.	O	O	O	O	O
I automatically did the right thing to achieve my goals.	O	O	O	O	O
Using the selection technique was fascinating.	O	O	O	O	O
Using the selection technique is fun.	O	O	O	O	O

Additonal notes:

Postquestionnaire

Rate the following statements:	Fully disagree	Mainly disagree	Neutral	Mainly Agree	Fully Agree
I felt comfortable pointing at objects with a controller.	O	O	O	O	O
I had enough time to practice.	O	O	O	O	O
It was easy for me to point at objects with a controller.	O	O	O	O	O
It was always clear to me, what I had to do.	O	O	O	O	O
It was always clear to me, how to use the controller.	O	O	O	O	O

Sort the selection technqiues by popularity (Names are written on the previous pages, you can also just use the given numbers on those pages):

1.	
2.	
3.	
4.	
5.	

Additonal notes: