Virtual Reality Project Report
Exploring the Effects of Movement on Object Selection in Dense 3D Virtual Environments - Group 2

Andreia Matos, Daniel Ramalho, Inês Correia, Lisa Grennberg

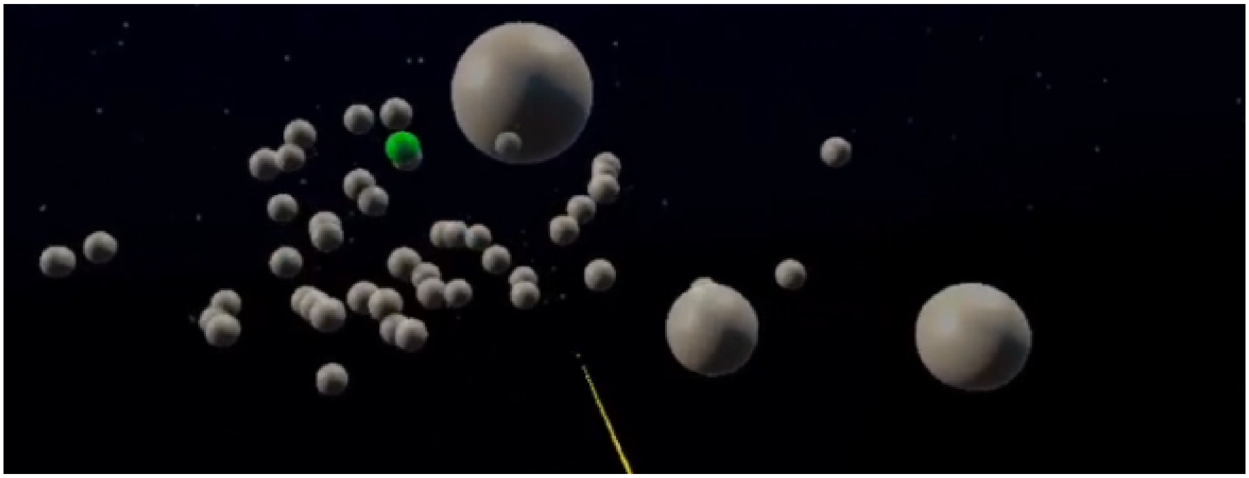


Fig. 1: The experiment environment.

Abstract—Target selection is a fundamental aspect of interaction. This is also the case for 3D environments, despite the research being more limited. We propose two selection methods: the Depth Ray and the Iterator Cursor, which derive from the ray casting metaphor. We adapt them to dense environments, occluded and moving targets. Then we conduct user tests and comparatively analyze their performances. Results favored the ray casting metaphor, especially with the addition of movement.

Index Terms—Virtual Reality, Selection Techniques, 3D Interaction

1 Introduction

Virtual Reality (VR), nowadays, completely blends in with our daily lives. It has been declared as one of the technologies with the highest projected potential for growth as it has been developing at a very fast rate, being used in more and more fields every day. Particularly due to the worldwide pandemic we are currently undergoing, the need for VR is only being highlighted as a possibility to reach ends that under such situations are otherwise unfeasible [11].

Despite the continuous breakthroughs in research regarding various of the VR techniques, some of them have yet to be fine-tuned to this new format. This is the case of object selection in 3D virtual environments which is integral to most 3D applications. 3D selection does not conform to the standard 2D mouse metaphor as the targets have 3D coordinates therefore, it is crucial to develop new selection techniques better adapted to the circumstances at hand.

Of the various object selection techniques introduced by VR researchers as an attempt to solve this problem, the two most common are **hand extension techniques** - which involve selection by a 3D virtual cursor which has the coordinates of the hand or handheld input device, and **ray casting techniques** - which involve selection by intersection of the targets with a virtual ray with varying depth.

Although several versions of these selection methods have been developed, there are still many aspects that could be explored in more depth . In this paper we will be considering three of these.

The first is the **density of targets**. Particularly in ray casting techniques, this can affect performance by introducing ambiguity into the selection, as more than one object can be intersected by the ray's selection area.

The second aspect is **visibility of the goal target**. Sometimes the target objects may be occluded by other objects resulting in their invisibility. Some recurrent solutions to this have been viewing mode switching or the rotation of the scene, which are time consuming steps that could potentially be avoided.

The third aspect is **movement of the objects in the scene**. This introduces more complexity to the environment which could diminish the user's accurate usage of the selection techniques. Also, movement introduces a quick and possibly unpredictable alteration of the target's coordinates, which can also affect the performance.

We propose two selection techniques, the **Iterator Cursor** and the **Depth Ray**. We present their design and then, a user study conducted to compare their performances. The iterator cursor performed better overall, however the performance of both methods saw significant variation as the environment parameters changed.

2 Related Work

As was already mentioned, selection is one of the pillars of interaction in VR. Many methods of selection have been developed as an attempt to

- *Andreia Matos is with IST. E-mail: andreianmatos@tecnico.ulisboa.pt*
- *Daniel Ramalho is with IST. E-mail: daniel.ramalho@tecnico.ulisboa.pt*
- *Inês Correia is with IST. E-mail: inesmargarida1618@tecnico.ulisboa.pt*
- *Lisa Grennberg is with Chalmers. E-mail: lisagr@net.chalmers.se*

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

improve this pivotal task. In this section we will go over some methods for selection in 3D environments, developed by researchers in the field. We will then go over some techniques that focus on selection in dense environments, followed by some that consider occluded targets and lastly, some regarding the movement of the scene objects.

2.1 3D Selection Techniques

There have been multiple takes on the ray casting metaphor, from the early Liang and Green [10] "laser gun" selection, later extended to "spotlight selection" as the selection area was substituted for a conic area, to the aperture based selection [5] and 2D image plane selection [13]. A recurrent alternative is the hand extension metaphor, which Mine states could be used to select an object, in local interactions, involving a direct mapping from the user's hand to a 3D virtual cursor [12]; for distant objects nonlinear mappings are applied, permitting the user to control the depth of the cursor. This metaphor can result in longer selection times as it depends on the three dimensions. Zhai et al. [19] presented the silk cursor, which is a semi-transparent 3D volume cursor, which possesses a larger activation area but fails to adapt well to dense environments. There have been studies that further developed characteristics of the techniques mentioned above as to adapt them to more situations, such as Vanacken, Grossman and Coninx [18] have done, developing more complex versions of the Depth Ray and Bubble Cursor, which we have taken as inspiration for this work. There has also been very promising progress made in alternative selection techniques. There are now various forms of gaze-assisted selection and head movement based input, for instance Outline Pursuits [14] for the former and SmoothMoves [1] for the latter. We now discuss selection techniques which particularly focus on dense target environments.

2.2 Selection Techniques for Dense Environments

Ray casting techniques have to deal with the challenge of only being able to select the first intersected object. There have been proposed metrics such as Liang and Green's spotlight selection [10] as well as some further extension, Shadow Cone Selection [16] and some others [15], despite not formally tested. On another note, there are also ways in which the user can explicitly choose one of the targets within the selection range [9], for example a cycle through method as developed by Grossman et al. [7]. Various designs were compared which concluded that the best performance was by the Depth Ray. This method consists of moving a marker across the ray, along with forward and backward movements of the hand; the target selected is the one closest to the marker. This method was then tested in a denser environment by Vanacken, Grossman and Coninx [18]. In this study they augment to 3D and test another method, the bubble cursor, an area cursor that dynamically changes its size to always capture only the closest target, which has previously only been studied in 2D environments. We will take on both the Depth Ray and the 3D Bubble Cursor and further adapt and test them on various dense environments.

2.3 Selection Techniques for Occluded Targets

Regarding complete occlusion of the target object, little research has been done on integrating a solution into the selection techniques. Usually, methods such as rotating the scene or switching viewing modes are used as a solution to this issue. However, Vanacken, Grossman and Coninx [18] proposed rendering the cursor as semi-transparent similarly to the previously mentioned silk cursor [19] and using visual feedback to provide selection information. We will further discuss this mechanism later within each of the developed techniques.

2.4 Selection Techniques for Moving Targets

Considering the selection of moving targets, there was very limited research to find, particularly in 3D environments. We could, however, find some research on 2D environments, Hold [8] is a moving target selection mechanism that suggests temporarily pausing the content while the selection is in progress to provide a static target, which proved to outperform other traditional approaches. Nonetheless, pausing to select might in some applications undermine the point of the selection, besides creating an extra step and consuming time resources. We will test both the Iterator Cursor as well as the Depth Cursor on their performance in environments with moving targets, hoping to improve understanding on the matter.

3 System Design

Based on our research of previous work on selection in 3D virtual environments we gathered some principles to have in account when designing our own selection techniques. In this section, we are going to explain the design guidelines we followed and our design process.

3.1 Design Guidelines

Most methods available already respond to some of the requirements needed to create useful selection techniques. However, they don't completely satisfy when it comes to their performance under dense environments or in environments with a lack of visibility of the goal target. In addition, we think that the influence of having non-static objects in virtual environments is not very well understood in terms of 3D selection, so we intend to design a method that accounts for this.

As a result, we collected a list of design guidelines from previous work and added a new one to include the non-static objects challenge:

- Allow for fast selections
- Allow for accurate selections
- Be easy to understand and use
- Produce low levels of fatigue
- Satisfy the above for sparse and dense target environments
- Support selections for both visible and occluded targets
- Support selections for moving objects

3.2 Designing for Dense Environments and Occluded Targets

Frequently, in Virtual Reality applications it is required to select objects that are not completely visible from the user viewpoint, thus making the task much more difficult or even impossible if the goal target is located completely behind another object. Highly dense environments make more probable the occurrence of complete and partial occlusions, since there is more chance of objects obstructing other objects.
To overcome the problem of visibility we make the objects in the scene slightly transparent at the time of selection. This allows the user to see the target even if it is occluded, since we wanted to focus on selecting objects, and not on finding them.
This resolves one problem, but with a basic ray casting technique, some objects would still be completely occluded and thus impossible to select without moving the camera. Our strategy to resolve this problem is to have all the objects in the direction the user is pointing activated. Then, we include mechanisms that permit disambiguation between the target and the other objects activated. This, however, means that the actual selection requires separate steps on the behalf of the user.

3.3 Designing for Non-static Objects

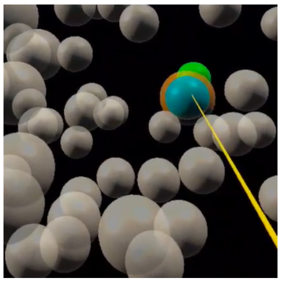
In actual virtual reality environments, it is common that the objects in the scene have some kind of movement. Therefore a selection technique should allow for efficient selections of both static and non-static objects. If a technique is not fluid and easy to use, it can become hard to catch an object that is constantly changing its position. For that reason, it is essential that the user finds the technique intuitive and precise.

4 System Implementation - Selection Techniques

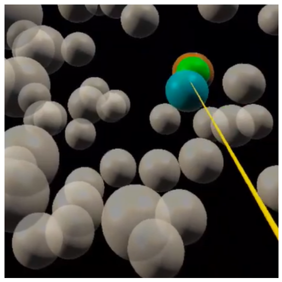
In this section, we will explain the selection techniques we implemented that should satisfy the design guidelines mentioned above. We came up with two different methods, one called Iterator Cursor and another called Depth Ray Cursor.

4.1 Iterator Cursor

Our first design is an extension of the ray casting metaphor.
For this method we have an yellow ray controlled by the user's hand. For the ray we used a long and thin cylinder.
The objects intersected by the ray are considered to be activated and change color to cyan. One of these objects is highlighted orange to show that it is the one that currently can be selected. If that object is the user's target all that's needed is simply to press the trigger button to select it. However, if the desired goal is not that first object but it also exists along the ray, the user can iterate between the activated objects to get to the goal target. The iteration is activated by clicking a designated button on the controller.



(a) User points in the direction of the target colored green. All the other objects in that direction change color to cyan.

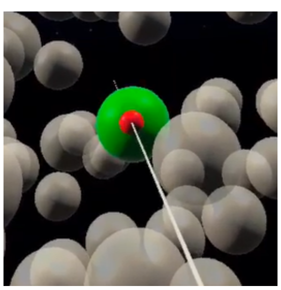


(b) User iterates between the objects in that direction to get to the green target. The object that can currently be selected is highlighted orange.

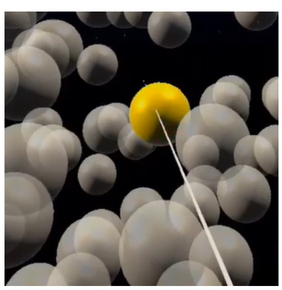
Fig. 2: This pictures display an use of the Iterator Cursor

4.2 Depth-Ray Cursor

For our second design we also used an extension of the ray casting metaphor.
In this method a white ray is cast from the user's hand. The user controls and uses it to point in the direction of his target object. We implemented the ray has a long thin cylinder.
A red sphere that we call Depth Marker exists and moves along the ray. This sphere is controlled by the user using the controller's joystick. Pressing the joystick forward makes the Depth Marker move further away from the user's hand and pressing backwards has the opposite effect. The user could also control the speed by aiming the joystick to the sides. To select an object, it is necessary that the Depth Marker intersects the goal target. When this happens the object changes color to yellow in order for the user to have a visual signal that this particular target is selectable.



(a) The user points in the direction of the green target and moves the red Depth Marker closer to it.



(b) When the red Depth Marker is in contact with the target, the target changes color to yellow indicating to the user that he can select.

Fig. 3: This pictures display an use of the Depth Ray

In both techniques, we send an haptic signal to the controller when an object is selected.

4.3 Milestones

This project was made for a University Course at Instituto Superior Técnico within the subject of Virtual Reality, and in order to be in tune with its evaluation methods, we divided the project into three milestones plus a final presentation.

- **Milestone 1:** Our initial goal for this milestone was implementing the two object selection techniques and also to create a simple environment with static objects. In fact, we were not able to complete all we have set out to do because none of the group members had previous experience working with the development environment or with Oculus Quest equipment. We also misjudged the difficulty of implementing the selection techniques. In the end, we were able to have a basic environment and to be more acquainted with the development program and the equipment.
- **Milestone 2:** In this milestone we were able to catch up for the lost time in milestone 1. We concluded the implementation of the two selection techniques and created an environment with objects to select. The plan for this milestone also included adding the movement dimension to the scene, and we were able to complete that as well.
- **Milestone 3:** Our aim for milestone three was to conduct user tests. We created various test stages to evaluate the performance of our selection methods, one stage with static objects and others with moving objects. The user tests we conducted are explained in further detail in the next section.
- **Final Presentation:** For the final presentation we gathered and analyzed all the data collected from the user tests and tried to draw some conclusion to our project. The data analysis is presented in detail under the section result.

5 User Study

The study presented in this paper is inspired by the previous work done by Vanacken, Grossman, and Coninx [18], where they conducted a study to test how well two specific selection methods performed when used to select targets within a dense 3D environment. This previous work showed that the Depth Ray selection method allowed efficient selections of objects in a dense environment where the objects would occlude one another. In this paper, they also performed tests on another method called the Bubble Cursor. This method became an inspiration for us when creating the Iterator Cursor used in our study. We understood early in the process that this selection method needed alteration to be able to function in our research study because the original Bubble Cursor is dependent on the objects being within close range of the user, as it is limited by the length with which the user is able to reach with his own arm holding the controller. This is the result of the Bubble Cursor not being based on the ray casting functionality as we decided to base the Iterator Cursor on, as explained earlier.

The major new thing that this paper brings to the research field is evidence of whether the Depth-Ray Cursor and the modified Bubble Cursor, called the Iterator Cursor also prove to be efficient selection methods for moving objects in dense environments or not. In this section, we present a user study where we performed tests to evaluate these two techniques, experimenting with environment density, movement speed, and the distance between the user and the targets. The goal of these experiments was firstly to evaluate how these parameters would impact the effectiveness of the selection for both methods separately. The second goal was to be able to compare the result from the performance of the two methods against each other to possibly find one to be more suitable. Lastly, we also had intentions to compare our results of experimenting with the selection of moving objects in a dense environment against previous work done on the selection of static objects in dense environments, if possible, to distinguish similarities and differences.

Our hypothesis about how the two selection methods will perform before conducting any of the tests is that the Depth-Ray Cursor will be more precise for target selection in dense environments where a large number of objects are occluded. On the other hand, we also hypothesize that the Iterator Cursor will be better suitable for the selection of moving objects than the Depth-Ray Cursor because it seems more adaptable in its position adjustment. Regarding which parameter will have the greatest impact on the efficiency of the two selection methods we predict it to be the increase of movement, meaning the speed in which the objects translate back and forth within the scene. We also believe the distance will have a negative impact on the efficiency.

5.1 Equipment

The equipment that we used to perform the user study was an Oculus quest headset and controller connected to a laptop. The advantages of using this equipment are that it is portable and does not require the tests to be done at one specific location. This is because the Oculus quest is an all-in-one VR headset where the tracking is compiled within the headset itself, instead of having to install separate motion sensors to map the user's movement in the test environment [3]. The Oculus quest equipment contains two 6DOF handheld controllers which can be represented by virtual hands in the scene that the player uses to interact with the virtual content [2]. The controllers can be tracked both in regard to position and orientation and this makes the interaction very precise. In this user study, we only made use of one of the controllers thus it was enough to perform the tests because the user only needed to use one hand.

5.2 Environment

The virtual environment that acted as the test scene for this user study was created in the program Unity. Unity is a real-time development platform where it is possible to develop VR and AR experiences or projects applicable to various industries [17]. The scene created for this research study was very simple in order to not distract the player from the main task of target selection. The overall scene was made dark blue and black while the objects got a light gray color to stand in clear contrast against the surrounding environment. These objects were not made completely opaque to allow the user to find the target even if it was occluded. The objects that were placed within the scene were all made in the shape of a sphere with a radius of 0.4m. We choose to not create objects with different sizes because that would have introduced another variable to the test which we did not want. A function was then implemented to generate a given number of spheres within the scene where the distance between the user and the object could vary between two specified variables, minimum distance, and maximum distance. Regarding how the objects moved in the scene we created five different movement variations that the objects could possess using the function Mathf.PingPong in the Unity script. Movement One only translated back and forth along the x-axis. Movement Two was set to translate both along the x-axis as well as between coordinates on the y-axis but with different sized variables. Movement Three was set to translate the object between coordinates on all three axes (x, y, z), also this time using different sized variables, in differentiation from Movement Four where the object also translated between coordinates on three-axis but with the same sized variables making it move along a straight diagonal line. Lastly, Movement Five was created to make an object move in a circle in front of the player. When the number of generated objects in the scene was specified it also at the same time got defined how many of the total generated objects would get assigned each movement variation. The test environment is displayed in Figure 4.

5.3 Participants

The tests in this user study were conducted with a sample size of 15 people, consisting mostly of other students, colleagues of ours, and family members. It would have been ideal to conduct the study with a larger number of participants to ensure more reliable results, but ethical and health concerns, in addition to a curfew and restrictions to movement imposed due to the ongoing pandemic severely impacted our ability to do so. The participants consisted of both males and females between the age of 20 to 27 years old with varying previous experience of virtual reality.

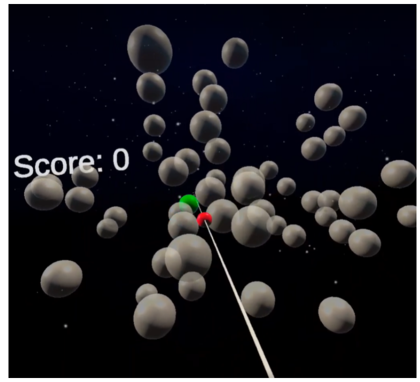


Fig. 4: Picture displaying the test environment created in Unity

5.4 Procedure

The user tests were given a "gamified" structure where the user will face different stages using each of the two methods on all of them. First, the user will go through the four stages using the Iterator Cursor to select target objects and then doing it all over again, but this time using the Depth-Ray Cursor instead. In each stage the user will be asked to select as many target objects as possible in the given time of 30 seconds, where each target is counted as one point in a counter displayed to the user in the scene. While the user performs the test, all trigger button clicks will be counted to then be compared with the score to detect "errors", meaning how many times the user took the select action without successfully selecting a target object. In each test stage, the target object that the user is meant to select will appear one at a time and the next target can only be identified after the previous one has been selected and thus automatically removed from the scene. The target object can be distinguished from the other spheres in the scene by changing to the color green, indicating to the user that this is the object they should strive to select. Before the test stages, users were given some time in a "pre-test" stage where they could learn the controls, experiment with the methods, and adapt to the virtual reality setting, without the pressure of timing or score.

5.4.1 Stage attributes

Four different stages were created to test the effects of **varying speed** and **distance** when selecting occluded objects in a dense environment. To learn about the exact values used for speed, distance and number of objects in each stage see Table 1.

- **Stage 1:** A static stage where the objects did not move. This static stage will serve as a base-line in evaluating the effects of later variations in movement and distance on the selection methods. The number of objects populating the scene was set to 55 and this parameter would remain constant during all stages.

- **Stage 2:** This stage introduces moving objects. The objects are now moving with a speed of 0.5 m/s with the five different movement variations presented earlier, but the distance between the objects and the user has not changed from Stage 1.

- **Stage 3:** In Stage 3 the objects will still move at the same speed as Stage 2, but they will be placed at a larger distance from the user. Here the objects will appear at a distance between 24–34 m instead of 4–10 m as they did in the previous two stages.

- **Stage 4:** In this stage, the objects will once again appear at the normal distance from the user, but the speed will be increased from the previous 0.5 m/s to 5.0 m/s.

Stage	Distance (m)	Speed (m/s)	Nr of objects
1	[4, 10]	0.0	55
2	[4, 10]	0.5	55
3	[24, 34]	0.5	55
4	[4, 10]	5.0	55

Table 1: Variable definitions for test stages

5.4.2 Breaking Point

After completing user tests in the four test stages, mentioned in the previous section, we realized that the Iterator Cursor seemed to consistently outperform the Depth-Ray Cursor, permitting users to collect a higher score, although at the same time it generated more errors. We were left somewhat unsatisfied, but also very curious. We still believed the Depth-Ray Cursor had some merit in its precision, and this made us question if, in the right circumstances that played more into the advantages of this method, we could see a significant increase in its performance, reaching a ”breaking point” where the Depth-Ray Cursor would surpass the Iterator Cursor's performance. These tests were not part of the original plan for the user study and were therefore not conducted with the same number of participants as the main study. The members of this group decided to test this new hypothesis among themselves with the knowledge that results indicated would require more extensive testing to be validated. To start with, we decided to double the number of objects populating the scene, to then systematical move up from there. The variables for speed and distance between the scene objects and the user would remain constant throughout all stages. To find the exact variables for each of the four Breaking Point stages, hereby referred to as Stage 1B to 4B, see Table 2.

Stage	Distance (m)	Speed (m/s)	Nr of objects
1B	[4, 10]	0.5	110
2B	[4, 10]	0.5	150
3B	[4, 10]	0.5	220
4B	[4, 10]	0.5	300

Table 2: Variable definitions for Breaking Point tests

5.5 Questionnaire

After participating in this user study all participants were asked to answer a questionnaire with questions formulated to capture users' thoughts and feelings regarding the efficiency of the two selection methods. The questionnaire consisted of a total of seventeen questions. In the first question, we asked about the users' familiarity with VR and the last question was a non-mandatory one asking if the users had any suggestion of possible improvements for any of the two selection methods to make them more efficient in the selection of moving objects.

The rest of the questions handled how the users experienced that the various test parameters in the different test stages affected the two selection methods, and the users were asked the same five questions two times concerning one selection method at a time. When answering the questions, the users were presented with a proposition and then had to select one number between one to five depending on if they agreed or not. On this scale, the number five equaled strongly agree, and one equaled strongly disagree, leaving the number three to become a neutral choice. Weighted answered of this kind makes it easy for researchers in a later step to add up the numbers from a specific question and compare two test objects against each other, which was ideal in this case.

Lastly, we also asked the users which method they preferred to use for the selection of moving objects in dense environments.

5.6 Results

In this section we will be analyzing the results obtained in the user study, starting with the measures of User Score and Error Rate, and then looking at the questionnaire. For the sake of brevity, the names Iterator Cursor and Depth-Ray Cursor will be abbreviated to IC and DRC, respectively.

5.6.1 User Score

Stages 1 - 4

It was somewhat expected that DRC would achieve lesser scores due to the time required to adjust the cursor, as this method contains the added z-axis dimension, or depth, while IC does not. However, it was immediately observable that the gap in score between the two methods would surpass our expectations. The average score, on the baseline Stage 1, was 19.33 for IC and 9.07, less than half, for DRC. This difference would increase even further in later stages, as can be observed in Figure 5.

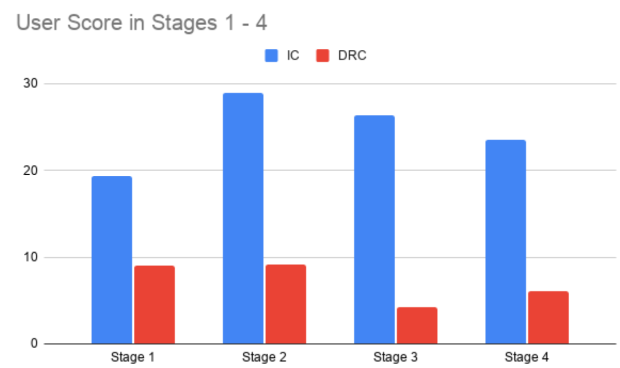


Fig. 5: Average number of points achieved by user in stages 1-4, for both selection methods

Taking a look at the distribution of the results, we see that the IC had a greater variance, at 13.42 compared to 8.33 for the DRC. However, somewhat unintuitively, both methods have the same distance in points between their lower quartile and higher quartile of results. The real difference can be observed in the outliers. While they present themselves in a similar distance from the respective medians, IC has both low (12) and high (27) outliers, while DRC shows only low (2, 3) outliers. This suggests that DRC is more difficult to excel at, but the sample size is too small to say this conclusively. Either way, these distributions did not point to a greater intuitiveness in the IC, as we believed they would. The next stages would generally see less agreement shown in the distributions, but the same overall trends.

The increase of IC score in Stage 2 was very evident, as an occluded object, by moving, would generally expose itself in the gaps between objects, allowing the user to point directly at it. The magnitude of the increase, however, reaching around 50% better scores, was surprising to us, and in the end the average score in Stage 2 (28,93) reached higher than the highest score in Stage 1 (27).

As for DRC, the change was negligible. It is not surprising that the gaps in occlusion did not benefit this method, as the user would still have to navigate the Depth Marker as before, but the absence of a negative impact is note-worthy, demonstrating that, by itself, slow movement did not pose an obstacle for DRC.

The same cannot be said for Stage 3, where DRC score was cut in less than half with the addition of distance, while IC score only fell with about 10%.

A decrease in score was to be expected when the distance was increased, and we can analyze the reasons for the decrease, and its particular impact on DRC, in accordance to Fitts's Law [4], which dictates that selection time increases with greater distance and smaller target size. The first phase of Fitts's selection task, consisting of a fast and imprecise movement towards the target [6], has a similar impact on both methods. The x-axis and y-axis movement is very similar, decreasing somewhat since objects appear closer together in the distance. The Depth Marker for DRC must be moved a greater distance, but this is a one time action as its position does not reset after selection. However, the greater impact was observed in the second stage of selection [6]: a slower and more precise movement in order to acquire the target. Both methods suffered from the smaller target size, but DRC's greatest issue was a loss of spatial perception, as the perceived decreased size of the Depth Marker meant the user could not always tell at a glance what the position of the target object was in relation to the cursor. This meant that the user would often not be sure if the Depth Marker should be moved backwards or forwards, provoking a stronger reliance on trial-and-error for the z-axis movement.

The fast movement in Stage 4 predictably decreased scores for both methods when compared to the previous slow movement. The more interesting observation for this stage, was that for the IC, score was still higher than in the static scene, meaning that the advantage of objects changing their occlusion was greater than the disadvantage of aiming at a fast moving object.

Stage	IC	DRC
1	19,33	9,07
2	28,93	9,13
3	26,4	4,2
4	23,47	6,07
Overall	24,5325	7,1175

Table 3: Average Scores for Stages 1-4

Looking at the overall results it becomes undisputed, as per Table 3, that in these stages, IC allowed for much greater scores, at an average 345% of what DRC achieved. Furthermore, we can observe that the static Stage 1 was without a doubt where the IC performed worse, having a difference of more than 20% with the next worst scoring stage.

Stages 1B - 4B

In the second phase of testing, increased occlusion showed somewhat different results, as per Figure 6.

Our hypothesis of reaching a "Breaking Point", where the number of objects meant less opportunities for the target object to move out of occlusion, and thus saw DRC become the highest scoring method, was not quite reached, but what we can observe is that the increase in occlusion did indeed translate into a steeper decline for IC than for DRC. In fact, comparing the averages from Stage 2 in the previous part of the study, to the averages of the scene with 300 objects, we can observe that IC fell to 44% of its original scores, while DRC fell only to 73%.

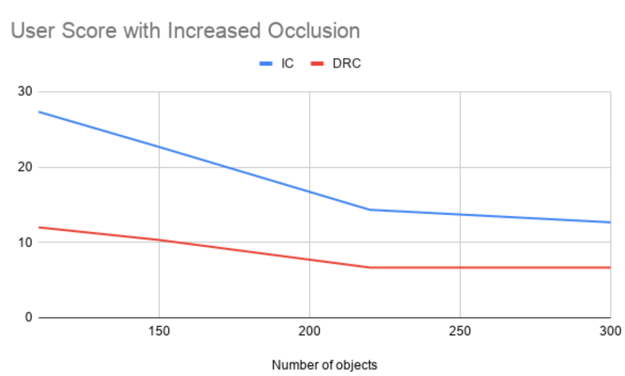


Fig. 6: Average score in stages 1B - 4B for both selection methods

5.6.2 Error Rate

Stages 1 - 4

The baseline observations that can be taken from Stage 1, as per Figure 7, are that DRC has less errors in absolute terms, but not relatively, when compared to score. As a matter of fact, when we calculate the % of total clicks that were "misses", or errors, DRC reached 30% , while IC only reached 20%, making it debatable if this method can be considered more precise.

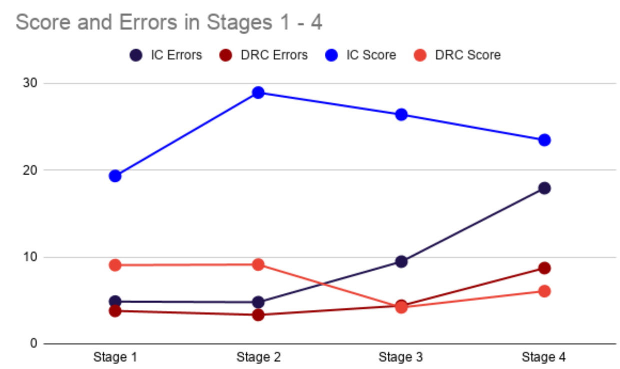


Fig. 7: Average Score and Errors for both methods in Stages 1 to 4

The introduction of movement in Stage 2 introduces only negligible changes, and the slight decrease for both methods is more likely related to the users becoming more familiarized with the methods.

However, the next stages see a significant increase for both methods. To exemplify, IC has 3.7 times more errors in the last stage, Stage 4, than in Stage 1. We attribute this huge increase to the fact that error correction is very quick in this method, combined with the lack of penalization for missing. This meant users did not have to make sure their first click hit the target, and some would naturally tend to click the select button multiple times for an object even if not necessary - so as to make sure that one of the clicks would hit the target. DRC had a smaller increase, of 2.6 times, and we believe that a longer time for error correction is linked to this, as the z-axis movement is slower and more subject to trial-and-error when under pressure. This meant the users tended to be more careful with their selection.

We were surprised by observing that, in Stages 3 and 4, the DRC score was surpassed by the number of errors. We conclude that the combination of having less score than IC, bigger error rate than IC, and now having more errors than score, leaves no doubt to the fact that DRC's expected precision, in controlling the z-axis manually, does not compensate for its shortcomings in these conditions, namely fast movement or distant objects.

Stages 1B - 4B

Both methods saw an overall trend of small, steady increase in errors during these stages. Some deviations were observed, namely a decrease in errors in Stage 2B for the IC, and a sharp increase in errors in Stage 3B for the DRC. However, due to the small sample size, and the fact that these deviations are not supported by the overall trend of the stages, these deviations are inconclusive at best, and as such we consider them to be of little note.

5.6.3 Subjective Feedback

From the data that we gathered from the questionnaire that all test participants were asked to answer after they had done the testing, we could make some conclusions regarding users' thoughts about the two selection methods. When comparing the answers, we saw that the users thought both methods were easy to understand but when it came to whether the methods were considered effective the response differed. The Iterator Cursor was considered to be a good selection method for choosing moving objects and users did not seem to think that it was essentially better for the selection of static objects. For the Depth-Ray Cursor on the other hand the majority of users did not consider it to be a good method for selecting moving objects and did not consider it to be much more effective for selecting static objects either. The majority of the results actually showed that the Depth-Ray cursor was not thought of as a good selection method in general, when considering the test goal of selecting as many objects as possible witin a given time. These result can be observed in further detail in Figure 8.

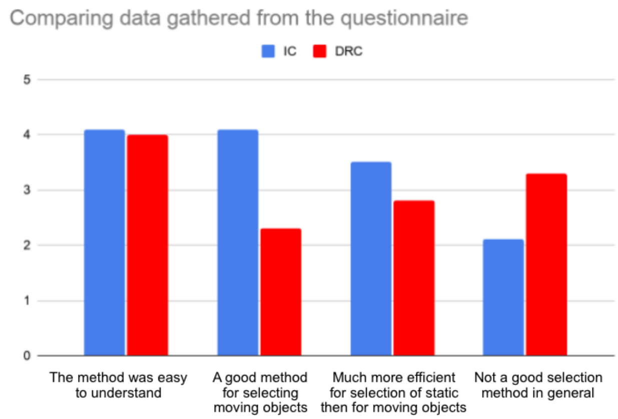


Fig. 8: Comparing answers from the questionnaire regarding the two selection methods. The questions was ranked from strongly disagree =1 and strongly agree =5.

User appreciation of the DRC was, in general, more affected by the variation in parameters that we tested when compared to the IC. According to the collected data, the users thought that the Iterator Cursor's effectiveness decreased the most when the objects moved with a greater speed (Stage 4), and the Depth-Ray Cursor was mostly affected by having objects appearing at a longer distance from the user (Stage 3), observations that are supported by the data. Users also remarked on the effect of the objects moving out of occlusion as being favorable, while they would often express dispraise of the error correction with DRC and its slower speed. This also became visible in the questionnaire when users answered the question of what they would like to re-design with the selection methods to improve their performance. Several participants answered that they would like to make improvements by "make it faster to adjust the Depth-Ray Cursor" and "improve the ease of the Depth Ray. Other suggested more specific improvements for the such as to make it easier to stop the cursor at a target when it's hovered or to make use of a larger selection radius to avoid going over the target. Regarding the Iterator Cursor, the only suggested improvement mentioned was to make the hovered object more clearly highlighted, so as to differentiate the target with greater ease. These results can be observed in further detail in Figure 9.

To conclude the data analysis from the questionnaire we learned that the majority of our test participants were not very familiar with VR from before, see Figure 10, which might have made scores lower, but it can also be considered an advantage that gave us a chance to see how intuitive and easy to learn these selection methods actually were. On the final question in the form users were asked which method they preferred for the selection of moving objects and 100% answered the Iterator Cursor. This gives us a clear indicator of which method is actually most suitable for this specific task performance of choosing moving objects in a dense environment.

6 Discussion and Future Work

We have analyzed results which provided further insight into these methods, that can have interesting implications for their practical usage, or that of similar implementations. Our results showed that movement can improve the performance of selection methods by providing gaps in occlusion. Furthermore, they indicated that methods requiring more

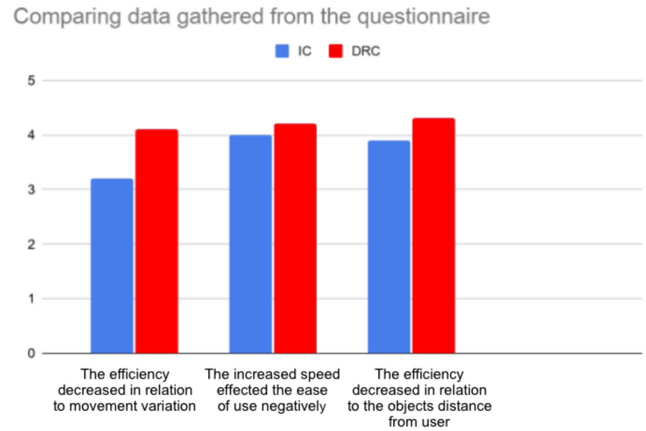


Fig. 9: Comparing answers from the questionnaire regarding how the changing variables affected the two selection methods efficiency. The questions was ranked from strongly disagree =1 and strongly agree =5.

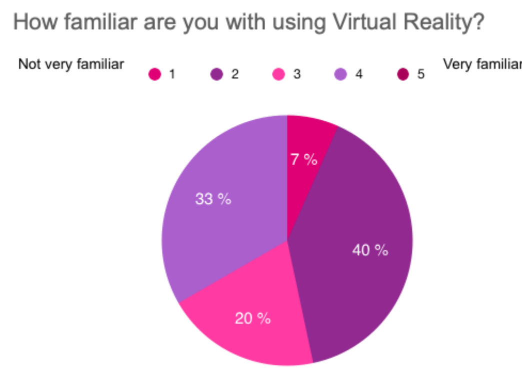


Fig. 10: Diagram displaying the distribution of the participants previous familiarity with VR.

visual feedback for the selection task (such as DRC's depth) can suffer greatly from distance to target. We believe the Depth-Ray can have its uses, especially in very high occlusion environments, but our results indicate that it is much more restricted by environmental variables such as speed and distance, when compared to a simpler method like our Iterator Cursor. In addition, we believe perhaps some modifications to the Depth-Ray Cursor, such as a larger Depth Marker, or some indication of its relative depth to the target, could aid its performance, although this would require further validation through testing, and the added complexity might separate it further from the Iterator Cursor in terms of intuitiveness for the user. Regarding the Breaking Point tests, we believe that in our test conditions, the hypothesized inversion would not be feasible, as with 300 objects there was still a notable distance in averages, and a further increase in number of objects would risk becoming absurd. Regardless, we would have liked to have access to a greater sample size to verify these findings, and we expect that the user feedback would have been more favorable to the DRC in these conditions. We also believe that attempting to clarify the limit where one type of method surpasses the other could make an interesting development.

7 Conclusion

In this study we have attempted to push the limits of our selection methods, and analyzed how they performed with different environmental conditions. We believe our observations have added perspective to the practical usage of these selection methods, for future use in virtual reality environments.

Acknowledgments

The authors wish to thank the experiment participants and Professor Augusto Esteves for his assistance in our implementation.

References

[1] A. Esteves, D. Verweij, L. Suraiya, R. Islam, Y. Lee, and I. Oakley. Smoothmoves: Smooth pursuits head movements for augmented reality. page 167–178, 2017.

[2] Facebook Technologies, LLC. Oculus device specifications – 3dof vs 6dof. https://developer.oculus.com/design/latest/concepts/oculus-device-specs/, 2021 (Retrieved January 12, 2021).

[3] Facebook Technologies, LLC. Underneath your expectations – unlock your pc´s power. https://www.oculus.com/quest-2/ , 2021 (Retrieved January 12, 2021).

[4] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement, 1954.

[5] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. pages 95–96, 1996.

[6] E. D. Graham and C. L. MacKenzie. Physical versus virtual pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 292–299, 1996.

[7] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi finger gestural interaction with 3d volumetric displays. pages 61–70, 2004.

[8] Hajri, Fels, Miller, and Ilich. Moving target selection in 2d graphical user interfaces. pages 141–161, 2011.

[9] K. Hinckley, R. Pausch, J. Goble, and N. Kassell. A survey of design issues in spatial input. pages 213–222, 1994.

[10] J. Liang and M. Green. A highly interactive 3d modeling system. computers and graphics. pages 499–506, 1994.

[11] R. Llamas and T. Mainelli. Worldwide augmented and virtual reality hardware forecast. https://www.idc.com/getdoc.jsp?containerId=US47225121, 2021.

[12] M. Mine. Virtual environment interaction tech-niques. 1995.

[13] J. Pierce, A. Forsberg, M. Conway, S. Hong, and R. Zeleznik. Image plane interaction techniques in 3d immersive environments. pages 95–96, 1996.

[14] L. Sidenmark, C. Clarke, X. Zhang, J. Phu, and H. Gellerson. Outline pursuits:gaze-assistedselection of occluded objects in virtual reality. 2020.

[15] A. Steed. Towards a general model for selection in virtual environments. pages 103–110, 2006.

[16] A. Steed and C. Parker. 3d selection strategiesfor head tracked and non-head tracked operation of spatiallyimmersive displays. international immersive projection technologyworkshop. 2004.

[17] Unity Technologies. Everything you need to succeed. https://unity.com, 2021 (Retrieved January 12, 2021).

[18] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *2007 IEEE Symposium on 3D User Interfaces*, 2007.

[19] S. Zhai, W. Buxton, and P. Milgram. The ”silkcursor”: Investigating transparency for 3d target acquisitio. pages 459–464, 1994.