

Dense and Dynamic 3D Selection for Game-Based Virtual Environments

Jeffrey Cashion, Chadwick Wingrave, and Joseph J. LaViola Jr., *Member, IEEE*



Fig. 1. User Study Configuration. A study participant just moments after correctly selecting an object in scenario 1.

Abstract—3D object selection is more demanding when, 1) objects densely surround the target object, 2) the target object is significantly occluded, and 3) when the target object is dynamically changing location. Most 3D selection techniques and guidelines were developed and tested on static or mostly sparse environments. In contrast, games tend to incorporate densely packed and dynamic objects as part of their typical interaction. With the increasing popularity of 3D selection in games using hand gestures or motion controllers, our current understanding of 3D selection needs revision. We present a study that compared four different selection techniques under five different scenarios based on varying object density and motion dynamics. We utilized two existing techniques, Raycasting and SQUAD, and developed two variations of them, Zoom and Expand, using iterative design. Our results indicate that while Raycasting and SQUAD both have weaknesses in terms of speed and accuracy in dense and dynamic environments, by making small modifications to them (i.e., flavoring), we can achieve significant performance increases.

Index Terms—Interaction techniques, game-based virtual environments, 3D object selection, dense and dynamic objects.

1 INTRODUCTION

Video game “play” and the rich environments of games are profoundly different from typical virtual environments (VEs) where the original guidelines for 3D selection were created [6]. Current video games have detailed, interactive scenes created with advanced modelling and animation software and rendered with hardware accelerated graphics and physics. Their interaction occurs with commodity 3D motion controllers and body-based sensing, similar and in some ways more advanced, than those found in traditional VEs. As such, guidelines for 3D selection are less relevant in these game-based VEs that routinely have dynamically moving and densely packed objects in the environment, either for realism or as

part of the gameplay. However, the exact moments where existing selection guidelines fail to be applicable are where much of the “fun” of the game can be impacted by bad selection.

The focus of our work is to revisit 3D selection for dense and dynamic game-based VEs by exploring the existing 3D selection guidelines and adding to them as appropriate. The criteria we are concerned with are mostly speed and accuracy but we are also looking for emergent criteria specific to these environments. Our basic approach takes two techniques optimized for the extremes of these requirements, evaluates their issues, and explores the design space between the two. Following the idea of “flavors” [2,17], we look for issues performing selections and find solutions to them through iterative design. We started with Raycasting [6] and SQUAD [9]. Raycasting is a commonly used 3D selection technique, where a virtual laser is projected into the world and selection is determined by either the collision or closeness of this ray to a target object. This technique is common, fast, and easily understood by its users, but is problematic for the selection of small or occluded objects. The SQUAD technique was designed for object selection in dense environments by creating a selectable list of objects through a conecast [10] and dividing these into groups of four; iteratively reducing this list by subsequent selections. While extremely accurate, it increases the number of selections and removes the

• Jeffrey A. Cashion is with University of Central Florida, E-Mail: jcashion@knights.ucf.edu.

• Chadwick Wingrave is with University of Central Florida, E-Mail: cwingrav@eecs.ucf.edu.

• Joseph J. LaViola Jr. is with University of Central Florida and Brown University, E-Mail: jjl@eecs.ucf.edu.

Manuscript received 15 September 2011; accepted 3 January 2012; posted online 4 March 2012; mailed on 27 February 2012.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

environmental context from the selection.

From these two techniques, we iteratively developed two 3D selection technique variations. The Zoom technique is an extension to Raycasting that helps deal with small or partially occluded objects by first zooming in on the region of potential targets. The Expand technique is a variation of Zoom and SQUAD that helps to deal with progressive refinement problems by placing the target objects in a grid for the user to choose from.

We then conducted a summative evaluation, comparing all four techniques across five different selection scenarios based on variations of object density and movement. These five different selection scenarios, to be more valuable to game developers, are modelled after ecologically valid situations, as opposed to constrained and controlled laboratory conditions. For instance, a fruit stand or cylinders as they would be stacked on a shelf as opposed to only floating spheres. This allows our evaluations to identify more realistic issues, at the cost of experimental control. From the results of our evaluation, we begin to develop new guidelines for 3D selection in dense and dynamic environments that can act as a complement to the existing guidelines of 3D selection [6]. We believe these new guidelines have the potential to assist game developers and designers who want to make use of 3D motion controllers for selection.

2 RELATED WORK

Selection is one of the universal interaction tasks [6] in 3DUIs and as such, has been extensively studied. While multiple guidelines have been created for developers of 3DUIs, the research has clearly shown that there is no best selection technique for all situations. Selection can vary greatly by specific tasks requirements, the environment's layout and the user populations' preferences and experiences, such as video game experience or cognitive traits [18,20]. Other factors such as fatigue, feedback, view of the environment, the follow-on manipulation technique or "fun" can also impact selection technique requirements. As such, we find that applying laboratory results to real-world selection tasks can lead to surprises.

Taxonomies [4] and computational approaches [14] have been used to explain and explore the design space of selection, to attempt to be predictive of performance. The taxonomic approach decomposes selection into the tasks of: indication of object and indication of selection and feedback; then identifies new methods of performing these tasks. The computational approach looks to create general computer models that encompass existing techniques and show opportunities for improvement. One weakness with both approaches is that they largely ignore individual user differences, preferences and strategies [11,17], which have been shown to greatly impact performance. This lack of focus on the user is problematic as evidence exists that users do not have a model of selection, but only of how to respond to what they are presented [19]. This is further supported by [18] which found users performing ray-casting selection increased angular error even though more time was spent on selections in order to increase visual feedback. So, despite efforts, no predictive or encompassing model of selection exists.

The best approach to optimizing selection is task analysis and iterative design [3]. This approach can greatly increase the usability of a technique by tailoring it for its specific optimization in the given application, domain, task, device, environment and user [3].

One methodology for adding optimizations to a technique, or "flavor" [2], is the iterative issue-solution map (IISM) methodology [17]. The IISM methodology creates a map of selection issues and their potential solutions based upon the literature, the designer's experience and short brainstorming sessions. Formative iterations uncover the positive and negative impact between issues and solutions, fleshing out the design space. This map of trade-offs allows the designer to create a more optimal technique through a clear understanding of the design space. This approach was used to generate flavored versions of ray-casting that improved the usability of the technique under specific conditions [17].

2.1 Dense and Dynamic Environments

In this research, we are looking at environments with densely clustered objects and dynamically moving objects. Both of these lead to more difficult selections. We imagine situations where users operate control panels or novel story-inspired interfaces or even interact more realistically in the environment such as grabbing small objects to throw, stack or even simply to kick. While dense selections have been heavily studied, dynamic selections have seen little attention (probably due to their rarity in traditional VR, where simulations can be paused). However, dynamic scenes are common in game environments, where the movement itself can be considered part of the "fun".

Most dense selection techniques operate by allowing novel methods for the user to reduce and control the selection space. So, while traditional ray-casting reduces dimensionality of a selection to a single ray, cone selection [10] allows users to define selection volumes and Aperture selection [5] provides them control of the volume's size and can use hand orientation to further reduce the selection dimensionality. Another approach is the flexible pointer [12] where Raycasting rays are bent by the user to point around occluding objects. One ray-casting flavor [17] pulls objects closer to the user for easier selection, based on the nearness to the ray, reminiscent of fish eye lenses.

Looking at the Bowman taxonomy [4], we see that feedback is another method to improve selection in dense environments. While feedback has been shown to slow down selection for simple selections [18], it can greatly improve user performance for difficult selections. The 3D Bubble Cursor [16], similar to the snap-to flavour [17], is an example of novel feedback, using a bubble that jumps between potentially selectable objects, to reduce a user's Gulf of Evaluation. Feedback can also be used to indicate the results of Steed's temporal aspects of selection, as used by the IntenSelect technique [7] which bends a ray-casting ray towards an object, based upon nearness over time. IntenSelect is also one of the only techniques designed for dynamically moving objects as well. Because it operates over time, the movement of the target object itself, with the user's ray following it, helps to disambiguate the selection from other objects.

3 3D SELECTION TECHNIQUES

Of the many possible selection techniques to examine in dense and dynamic environments, we chose to begin with Raycasting and SQUAD because they represent techniques that were designed across the spectrum of object density; Raycasting for sparse environments and SQUAD for dense environments. Using iterative design, we were then able to build two variants to these techniques, Zoom and Expand, which we felt would improve upon the original techniques.

3.1 Raycasting

Raycasting is a simple selection technique and acts as a baseline for our iterative design and summative evaluation. This technique is analogous to shooting a laser pointer out of the center of the input device into the screen. The first collision reported back to the interface is accepted as the object which the user was pointing at. This technique is highly precise, yet not always accurate. However, this is often implemented in game environments as an occlusion target on the screen that extends into the scene. This is because games are played on televisions and not encompassing VEs that can create a continuous ray.

Raycasting is fast but has problems with small or moving targets. In scenarios with fast-moving objects, the user might need to effectively chase the object around the screen with their cursor, just to get the center of the cursor over one of the pixels used for that object. This can hinder performance and is a good reason why we hypothesize that Raycasting should not be used in anything but the simplest scenarios.

3.2 SQUAD

SQUAD is a selection technique that uses progressive refinement for narrowing the choice of objects to select from [9]. This is done by presenting the objects contained within a sphere-cast and displaying them on the screen in quadrants. The user selects the quadrant which contains the desired object, and then the objects which were in the same quadrant are then used to fill the quadrants, in the same manner that the original objects were. Any objects that exist in a quadrant that is not selected are simply discarded if they were clones or returned to their original context and position.

The strong point of this technique is that it is great for selecting an object that is even slightly visible on the screen, regardless of occlusion. Once the progressive technique is started, it can be guaranteed that the user can select the desired object, assuming there is little to no ambiguity between different objects. With few objects, the technique's multiple steps pose little overhead. When density increases, the number of steps required to work through the technique can introduce significant delay, yet still retain the possibility of perfect accuracy.

The primary caveat with SQUAD is that it poses problems when there are multiple similar looking objects being displayed. This originates from the fact that the objects are removed from their original context and placed in the SQUAD quadrants. If a user wanted to select a particular instance of an object that exists alongside other instances, then there is virtually no way that the user can determine which one they want once the objects are placed in quadrants. It may be argued that if the objects are the same then it really should not matter that this weakness exists, but it still impedes a user's ability to properly select when objects are similar, even when not identical. The similarity could be color, modest shape deformation, or even simply a desire by the user to select front-most objects or back-most objects.

3.3 Zoom



Fig. 2. Zoom Method. Camera has zoomed in on objects.

To extend the basic idea of Raycasting, we designed a zooming technique that reduces the density of the objects by zooming in on them. For a given on-screen circular cursor, all objects which at least partially lie inside of the circumference of the cursor are considered potential targets. When the user makes their selection, the camera zooms in on the center (average) position of the potential targets. While the objects still maintain their relative position to each other, they now take up a larger percentage of the screen, thus providing the user with a larger area to aim at.

This technique does not solve the problem of occlusion. When zooming in, the exact same amount of occlusion will remain, as the camera field of view is narrowed. One slight enhancement of this technique is the hiding of all objects that were not initially inside the cursor when the user made their initial selection. For objects near the outer bounds of the cursor, they may now be exposed on their outer faces where they were previously blocked. Another challenge not solved with this technique is that of selecting a moving object. In

fact, it potentially makes it more difficult, since a smaller portion of the screen can be seen. If an object is moving and you zoom in on its position, it will more quickly move out of the camera's view, working against the original intention of the user. It is for this reason that we hypothesize that zooming should not be used in VEs with selectable moving objects.

3.4 Expand

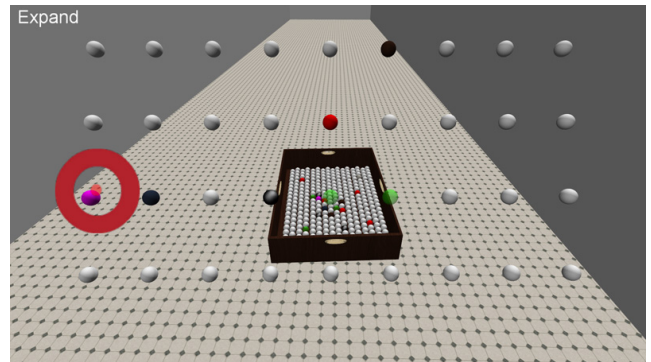


Fig. 3. Expand Method. Objects appear in grid.

The problem of losing original context was the primary drive for developing the Expand technique. When selecting, we felt that the objects should not just be brought into a secondary context and then have the user iteratively narrow down the choices. In our initial pilot studies, users often described this as tedious and too time consuming. The biggest concern is the problem of determining two or more similarly looking objects apart when brought into the secondary context. Any information about its original position is lost, making the selection process much more challenging.

Our technique was built in several steps. The design process we followed when creating Expand was based on the Iterative Issue-Solution Map [17]. We began by creating our own instance of the Raycasting technique. From this, we added the ability for the camera to zoom in on the potentially selected targets when a user attempts to select (i.e., the Zoom technique). This caused the objects of interest to occupy more of the screen, but did not take full advantage of the entire screen. It also established the technique as a two-step technique, introducing more time required to complete an entire selection. At this point, we envisioned Expand as an extension of Zoom with features and benefits of SQUAD but without the context problem.

The quadrant arrangement used in SQUAD was modified to be a virtual grid that filled the screen. The grid was dynamically arranged depending on the number of objects which needed to be placed in it, thus allowing the entire screen to be utilized. When the user makes their initial selection, only the objects with some part inside the circular cursor are brought forward to the grid. Objects that are not participating in this second selection step were made translucent to aid in clarity which was determined to be beneficial [13]. The act of transitioning the objects between their original position and their virtual grid position was controlled by the user via the input device. By moving the controller away from the screen and towards the body, the objects would transition to their grid position. By moving the controller forward towards the screen, the objects would transition back toward their original position. After some initial pilot studies, the controls were changed so that the transitioning of the objects to the virtual grid was done automatically once the user made their initial selection.

After initial testing was done with selecting moving objects, another iterative change was made. The original objects needed to be left in their original places so that the environment was not altered when making a selection, so rather than use the original objects when filling in the virtual grid, we cloned them and used the clones instead. This allowed us to do anything we wanted to the clones without worries that the originals were affected, which would have

had the potential to disturb other objects and cause unexpected side effects.

4 SUMMATIVE EVALUATION

In order to determine how well each selection technique performed in each of the different scenarios relating to object density and dynamics, we conducted a user study. Before conducting the study, we established several hypotheses:

[H1] Raycasting will be best suited to static, low density environments.

[H2] Zoom will be marginally better than Raycasting, overall.

[H3] Raycasting and Zoom will suffer in high density and dynamic environments. Likewise, SQUAD and Expand will perform considerably better than Raycasting and Zoom in these cases.

[H4] Expand will be at least marginally faster than SQUAD in both static and dynamic scenarios.

4.1 Subjects and Apparatus

We ran 28 participants (22 male, 6 female) with ages ranging 18 to 29 with a mean age of 21, recruited from the general population of the University of Central Florida. On average, participants played games about once a week, and half had previous experience using the Sony Move Controller. They rated their general gaming skill as average and felt comfortable using a controller to point to objects on a screen. The evaluation portion of the study lasted approximately 30 minutes, including the pre and post-questionnaire. Participants were compensated \$10 for their time.

Our experimental setup (see Figure 1) consisted of a Samsung 50" 1080P HDTV, a PC, and a PlayStation 3. The computer contained an Intel Core-i7 920 CPU with 8GB of RAM and an Nvidia GeForce GTX470 GPU. The Sony PlayStation 3 included the Sony Move.Me SDK and a PlayStation Move Motion Controller, an accurate 6 DOF tracking device that includes a set of buttons. The software used for testing was Unity 3.4, available from [15]. The study proctor and participant were the only two people in the room, and the setup was against a wall where there was minimal distraction.

4.2 Experimental Task

Participants were asked to test four selection techniques in five different scenarios. Both the order of the selection techniques and the scenarios were randomized. For each scenario and selection technique combination, the participant was given one minute to practice in a special practice scenario that was shielded off from the rest of the examination. The special practice scenario consisted of several medium sized objects which rotated about a central point at approximately 0.5 Hz. Participants could end the practice session at any time once they felt comfortable with the technique. Once completing the practice, the participant was notified that they had five seconds until the real testing would begin, and once this time was up, they proceeded to the first scenario. For each scenario, the participant was given two seconds to observe the scene and determine where the object to select was located. The target object was uniquely colored purple in the scene. Upon selecting an object, a note on the screen indicated a correct selection and they were given two seconds to transition to the next scenario. Audible feedback was also given to indicate when a selection was made. After all five scenarios were tested using each of the four selection techniques, the interactive portion of the study was complete.

When pointing the controller at the screen, the user was positioned approximately six feet away from the display. They were informed that the trigger button on the bottom of the controller was the only button they needed to press to perform a selection. Before starting the test, they were informed that if at any time during a multi-step selection process the desired object was not visible, to simply select an incorrect object and try again. The selection mechanism within the test was a 2D cursor controlled by the

PlayStation Move which projected directly into the VE with a ray for Raycasting or a cone for the other three techniques. This was done to avoid the problem of hand-eye mismatch [1].

4.3 Experimental Design and Procedure

We used a 4×5 within-subjects factorial design where the independent variables were selection technique and scenario. Selection technique varied between Raycasting, Zoom, SQUAD, and Expand. Scenario varied between the five scenarios described in Section 4.4. The dependent variables were task completion time and selection errors made. We also measured user preferences for each technique in terms of speed, accuracy, and usability, as well as asked them to rank the four techniques from 1 (most preferred) to 4 (least preferred).

Participants were first given a consent form and then briefed about its contents. Participants then were given a pre-questionnaire (see Table 1). Upon completing the pre-questionnaire, the participants were then brought over to the testing area where they performed the interactive portion of the study. The proctor coached them on how to use the Move Controller, as well as how to perform the selection in the testing environment. Once participants started the study, they were not interrupted or given any help. Once completed, participants were given a post-questionnaire (see Table 2). All of the questions except for Q8 were presented using a 7-point Likert scale where 1 was the most negative response and 7 was the most positive response.

Table 1: Pre-Questionnaire

Q1	About how often do you play video games?
Q2	What percent of your gaming is playing 3D games?
Q3	What % of your gaming is spent playing FPS games?
Q4	Do you have any experience using the Move Controller?
Q5	Where do you rank your general gaming skill?
Q6	How comfortable are you using a controller to point at the screen with?
Q7	Do you have any flexibility or pain issues with your primary hand or arm?

Table 2: Post-Questionnaire

Q1	How would you rate the Raycasting technique in usability? Speed? Accuracy?
Q2	How would you rate the Zoom technique in usability? Speed? Accuracy?
Q3	How would you rate the Expand technique in usability? Speed? Accuracy?
Q4	How would you rate the SQUAD technique in usability? Speed? Accuracy?
Q5	How adequate do you feel the 3D interface was?
Q6	How adequate do you feel the time allotted for practice was?
Q7	How comfortable were you using the Sony Move Controller?
Q8	Rank the selection techniques in order of desired use.
Q9	When determining how much you like using a selection technique, how much influence does <i>ease-of-use</i> have on your decision?
Q10	When determining how much you like using a selection technique, how much influence does <i>speed</i> have on your decision?

4.4 Scenarios Tested

We tested a variety of scenarios that encompass the spectrum of potential selection situations. These range from completely stationary and low density to high-speed moving objects with high density. These scenarios were designed to cover typical situations which might occur in a game-based VE.

4.4.1 Scenario 1: Medium Density, Medium Motion

Participants are presented with a large enclosed area which featured 40 floating cubes that move in a random manner with periodically changing directions (see Fig. 4). While the speed was not too high, the movement was unpredictable and thus the user was encouraged to focus carefully on tracking the object with the PlayStation Move controller. For any given moment, the cursor would have a modest amount of objects inside it, ranging from zero to around five. This low density suits SQUAD, as the recursive nature is kept shallow and thus is strongly similar to Expand with regard to time required to select.

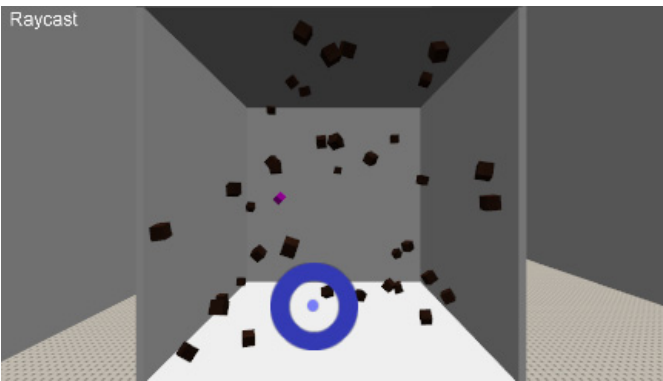


Fig. 4. Scenario 1. A user is presented with a box that contains many cubes which are moving in an unpredictable manner.

4.4.2 Scenario 2: High Density, High Speed

Participants are presented with a rectangular tray which featured 320 small spheres of varying color (see Fig. 5). The tray was rotating about the y-axis at approximately 0.5 Hz and the target object was off-centered, thus forcing the user to focus heavily on getting it inside the cursor. There were many colored balls near the target ball to enhance the difficulty of determining which ball was which.

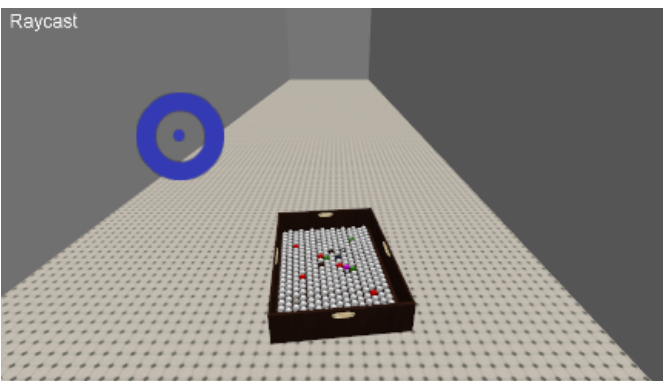


Fig. 5. Scenario 2. The user is presented with a rotating tray, filled with 320 small round balls. Color is used to heighten difficulty.

4.4.3 Scenario 3: Low Density, No Motion

Participants are presented with a fruit stand that featured several apples and bananas (see Fig. 6). The target object was a stationary apple that was off-center from the screen and required the user to

only move the cursor over to it. The apple was modest in size and relatively easy to select. Upon first inspection, the participant was expected to be distracted by the other container areas of the stand, which were actually textured and empty of selectable objects. Of all the scenarios, this was the most static and best suited for Raycasting.

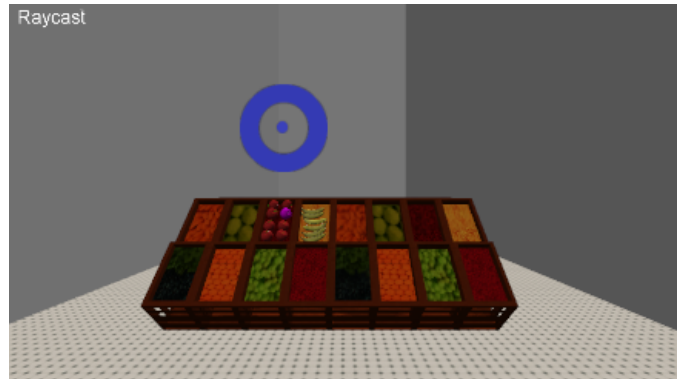


Fig. 6. Scenario 3. A fruit stand, which contains many sections filled only with a texture, but others with real 3D objects to select.

4.4.4 Scenario 4: High Density, Low Motion

Participants are presented with a rotating table that features 42 medium-sized boxes of varying color (see Fig. 7). The target object was a box which was positioned off-center, thus requiring the user to track it as the platform rotated. The motion is minimal, but enough to increase the difficulty beyond that of static selection.

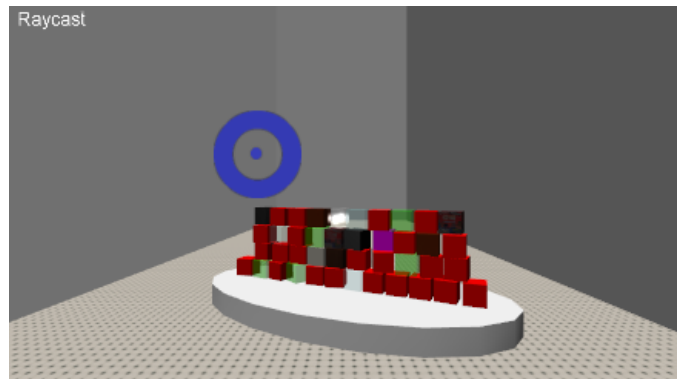


Fig. 7. Scenario 4. A rotating table with 42 boxes of varying color. The target object is off-center, thus requiring object tracking.

4.4.5 Scenario 5: High Density, No Motion

Participants are presented with a table top that featured sixty-six medium-sized cans (see Fig. 8). The target can was mid-way back and mostly occluded by neighboring cans. Participants were required to select the slim visible top portion of the can. This scenario emphasizes the difficulty in selecting a highly occluded object, even when it is stationary. This occlusion makes the selected area highly dense with the user having only a small area in which to hit the target.

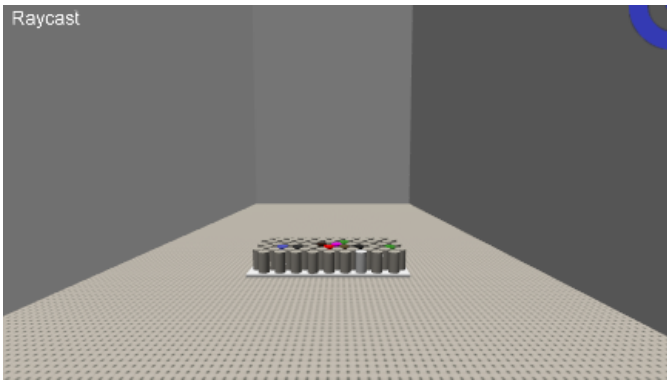


Fig. 8. Scenario 5. A tabletop featuring 66 cans. There is a lot of occlusion which obscures the target object, increasing difficulty.

4.5 Experiment Results

To analyze the quantitative data, we performed a repeated measures one way ANOVA on both completion time and number of errors made overall and for each scenario. When appropriate, we also ran post hoc analyses using t-tests. To control for the chance of Type I errors, a Holm’s sequential Bonferroni adjustment [8] with six comparisons at $\alpha = 0.05$ was used. Note that two outliers were detected for two participants in scenario 2 with completion times five standard deviations above the mean. Thus, we removed these participant’s data from the overall and scenario 2 analyses.

4.5.1 Overall

Fig. 9 and Fig. 10 show the overall mean completion times and average errors for each technique, respectively. We found significant differences for both completion time ($F_{3,23} = 6.4, p < 0.01$) and error rate ($F_{3,23} = 22.59, p < 0.001$) across the four techniques. Expand was significantly faster than SQUAD ($t_{25} = 4.64, p < 0.0083$) and Zoom ($t_{25} = -3.39, p < 0.01$), but not Raycasting, due to the Bonferroni adjustment ($t_{25} = 2.25, p = 0.03$). With regard to errors, Expand had significantly fewer errors than SQUAD ($t_{25} = 2.06, p < 0.05$), Zoom ($t_{25} = -5.56, p < 0.01$), and Raycasting ($t_{25} = -6.82, p < 0.0083$). SQUAD also had significantly fewer errors than Zoom ($t_{25} = -3.35, p < 0.0167$) and Raycasting ($t_{25} = -4.96, p < 0.01$).

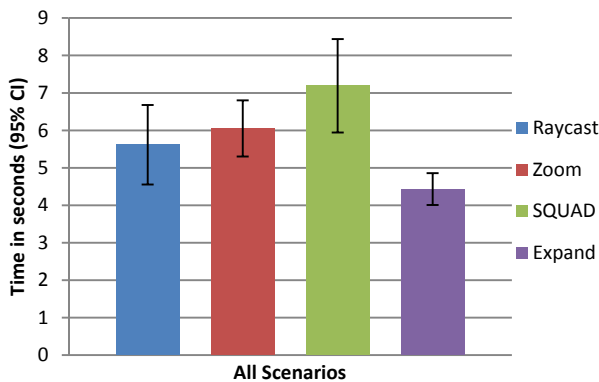


Fig. 9. Mean Total Time, All Scenarios. Expand is significantly faster than Zoom and SQUAD.

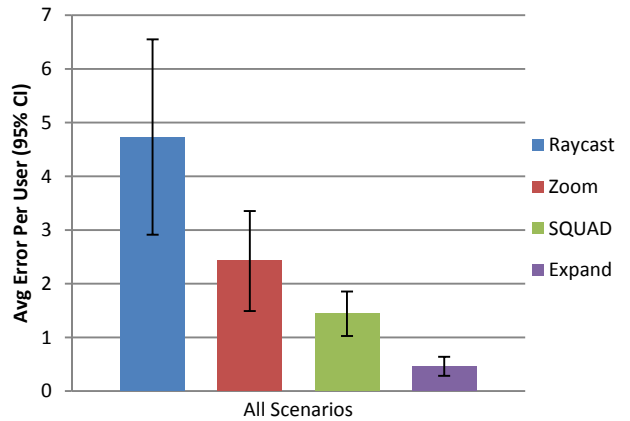


Fig. 10. Avg Errors, All Scenarios. Raycasting experienced the most errors, followed by Zoom, SQUAD, then Expand.

4.5.2 Scenario 1: Medium Density, Medium Motion

We found significant differences in mean completion time ($F_{3,25} = 3.70, p < 0.05$) for Scenario 1 (see Fig. 11), which had medium density and contained medium motion. In this scenario, SQUAD was significantly faster than Zoom ($t_{27} = -3.25, p < 0.01$) and Expand ($t_{27} = -3.71, p < 0.0083$) but not Raycasting due to the Bonferroni adjustment ($t_{27} = -2.51, p = 0.019$). There were no significant differences for errors ($F_{3,25} = 1.92, p = 0.133$) among the four techniques. The poor performance of Raycasting and Zoom can be expected since the difficulty in tracking a moving object is quite difficult, even if the velocity is modest. The SQUAD technique lets the user bring the moving objects to the forefront and make their selection from a group of non-moving clones. Since the moving objects were more spread out in this scenario, when the initial selection was made, SQUAD did not have many objects for the user to go through. This implies that SQUAD performs more like Expand in this case.

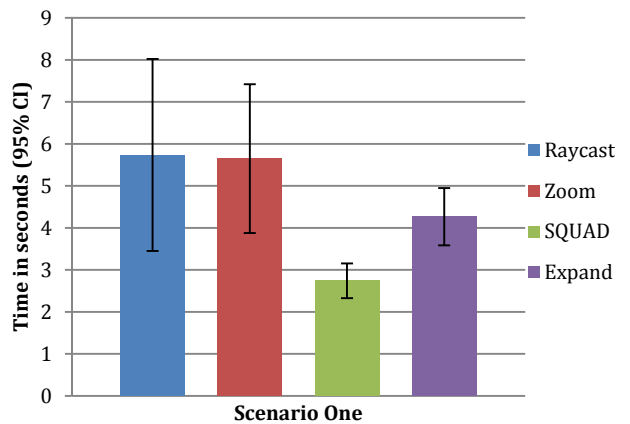


Fig. 11. Mean Completion Time, Scenario 1. SQUAD experienced the fewest number of errors, followed by Expand.

4.5.3 Scenario 2: High Density, High Speed

Scenario 2 featured a box with balls packed very densely. The box was rotating at a speed which made tracking the target object very difficult. Fig. 12 and Fig. 13 show the mean completion times and errors made for this scenario, respectively. Significant differences were found for both completion time ($F_{3,23} = 7.89, p < 0.001$) and errors made ($F_{3,23} = 14.14, p < 0.001$). The post-hoc analysis showed that Expand was significantly faster than SQUAD ($t_{25} = 4.49, p <$

0.00833), Zoom ($t_{25} = -3.52$, $p < 0.01$), and Raycasting ($t_{25} = -3.47$, $p < 0.0125$). Additionally, Expand had significantly fewer errors than Zoom ($t_{25} = -4.29$, $p < 0.01$) and Raycasting ($t_{25} = -5.60$, $p < 0.0083$) but not SQUAD, due to the Bonferroni adjustment ($t_{25} = 2.34$, $p = 0.028$). Although Expand and SQUAD are similar, in this scenario SQUAD suffered from the fact that the large number of objects increased the number of steps required to make a single selection. With each successive step, the user had to rescan all of the new quadrants to determine which contained the desired object. This greatly added to the total time required to make a selection. Both Zoom and Raycasting had relatively poor results in both speed and accuracy.

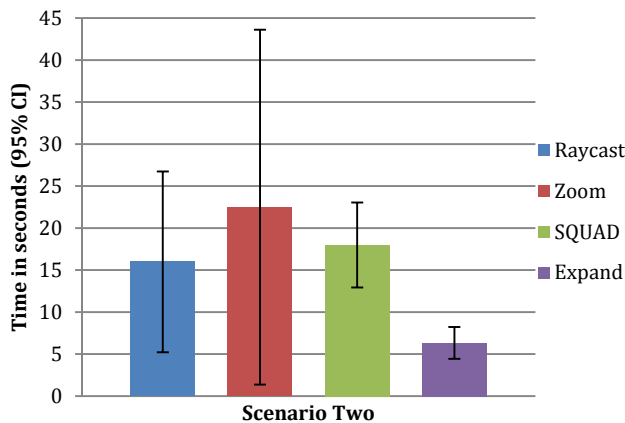


Fig. 12. Mean Completion Time, Scenario 2. Expand was significantly faster than SQUAD.

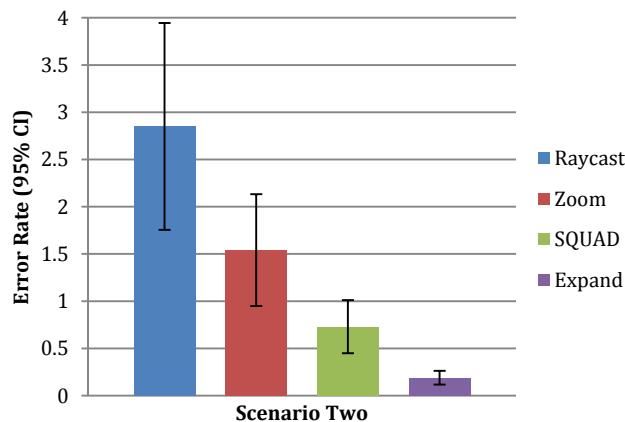


Fig. 13. Mean Errors, Scenario 2. SQUAD and Expand have significantly fewer errors than Raycasting, With Expand having significantly fewer errors than SQUAD.

4.5.4 Scenario 3: Low Density, No Motion

Scenario 3 featured stationary objects that were not very dense. Fig. 14 shows the mean completion times for this scenario. Significant differences were found for mean completion time ($F_{3,25} = 2.86$, $p < 0.05$) and since the objects were of ample size, Raycasting proved to be significantly faster than SQUAD ($t_{27} = 4.16$, $p < 0.0083$). There was also a trend toward significance for Raycasting over Expand ($t_{27} = 2.67$, $p = 0.013$) and Zoom ($t_{27} = 2.01$, $p = 0.046$). No significant differences for errors ($F_{3,25} = 2.24$, $p = 0.09$) were found among the four techniques in this scenario. The added overhead of the multi-step process for each selection added enough time to cause them to all take significantly more time than Raycasting, which provided a quick and easy way to point and select without any unnecessary extra steps.

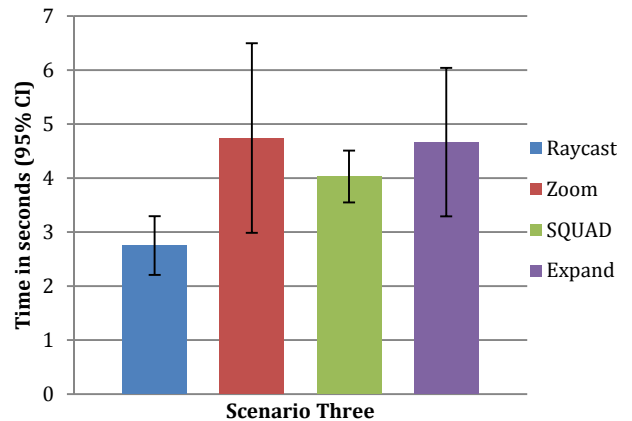


Fig. 14. Mean Completion Time, Scenario 3. Raycasting was significantly faster than SQUAD due to simple and direct selection.

4.5.5 Scenario 4: High Density, Low Motion

Scenario 4 featured several medium sized boxes which sat on a rotating platform. The rotational velocity was rather low, thus making it somewhat easy to select the target object. Fig. 15 shows the mean completion times for this scenario and significant differences were found ($F_{3,25} = 8.21$, $p < 0.001$). Despite the fact that the objects were moving, Raycasting was still significantly faster to use than SQUAD ($t_{27} = 3.54$, $p < 0.0125$), Expand ($t_{27} = 6.63$, $p < 0.0083$), and Zoom ($t_{27} = 5.60$, $p < 0.01$). No significant differences were found for errors made across the four techniques ($F_{3,25} = 1.46$, $p = 0.231$).

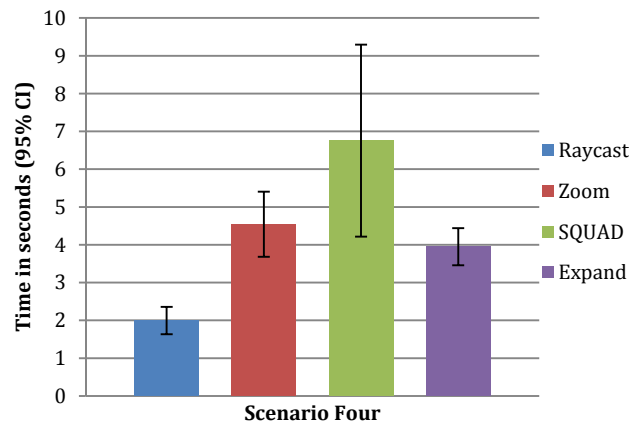


Fig. 15. Mean Completion Time, Scenario 4. Raycasting is significantly faster than all other techniques.

4.5.6 Scenario 5: High Density, No Motion

Scenario 5 featured a table with cans situated where the target object was mostly occluded from all sides. Fig. 16 and Fig. 17 show the mean completion times and total errors made for the four techniques, respectively. There was a significant difference for completion times in this scenario ($F_{3,25} = 2.75$, $p < 0.05$) but post hoc analysis did not reveal any further significance due to the Bonferroni adjustment. However, there were significant differences for errors made across techniques ($F_{3,25} = 15.12$, $p < 0.001$). Participants made significantly more errors with Raycasting than with SQUAD ($t_{27} = -4.43$, $p < 0.0083$), Expand ($t_{27} = -4.25$, $p < 0.01$) and Zoom ($t_{27} = -3.57$, $p < 0.0125$). This result can be likely attributed to the user rapidly re-attempting their selection when using Raycasting to make up for the difficulty associated with the low target object exposure.

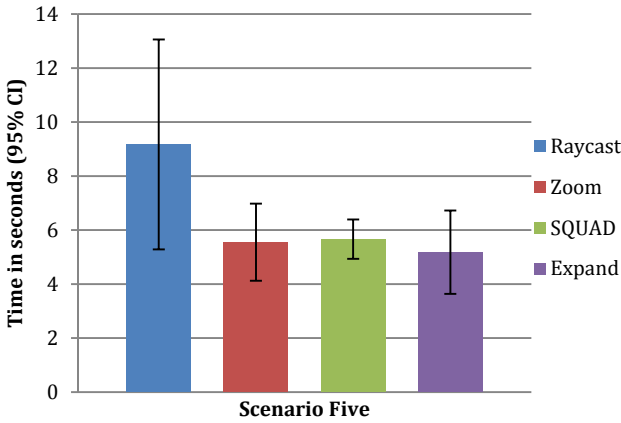


Fig. 16. Mean Completion Time, Scenario 5. Raycasting was slower than other three techniques, but significance was weak.

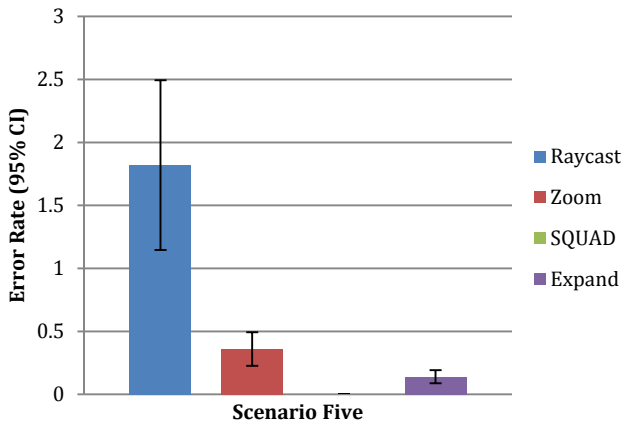


Fig. 17. Mean Errors, Scenario 5. Raycasting had significantly more errors than the other techniques. SQUAD experienced no errors.

4.5.7 Post-Questionnaire

We conducted Friedman tests on Q1-Q4 followed by Wilcoxon Signed Rank tests when appropriate. For each Wilcoxon Signed Rank test, six comparisons were made and Holm’s Sequential Bonferroni adjustment [8] was used at $\alpha = 0.05$ to control for the chance of Type-I errors. For Q8, a Chi-squared test was run to determine if there was a preference for any one of the techniques. Fig. 18 shows the mean ratings for Q1-Q4. For usability, significant differences were found across the four techniques ($\chi^2_3(N = 26) = 9.08, p < 0.05$). Specifically, participants rated Zoom significantly higher than Raycasting ($Z = -2.74, p < 0.0083$), while SQUAD ($Z = -2.02, p = 0.044$) and Expand ($Z = -2.33, p = 0.02$) were not rated higher than Raycasting due to the Bonferroni adjustment.

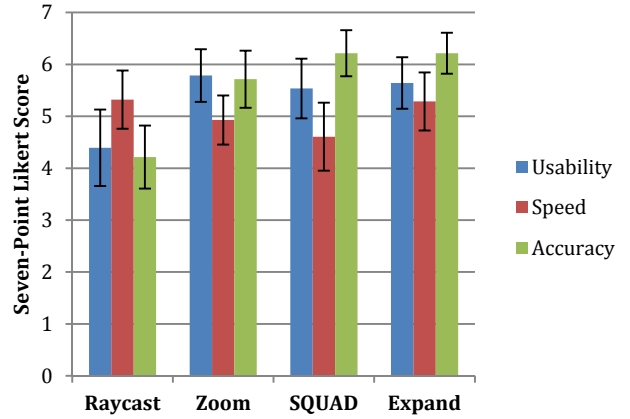


Fig. 18. Post-Questionnaire, Technique Critique. Raycasting was less usable and less accurate than other three techniques.

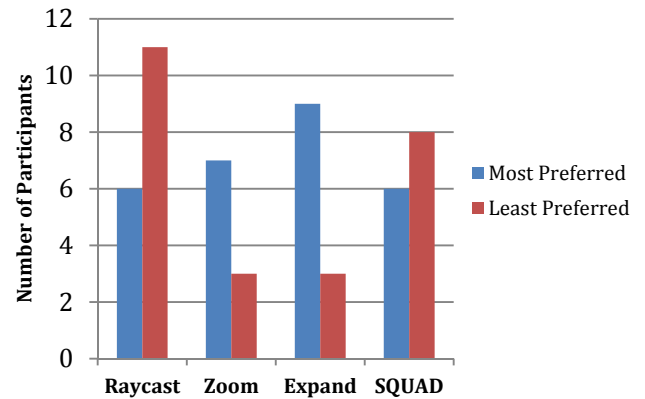


Fig. 19. Overall Ranking of Selection Technique. There was no significant favorite among all techniques.

Interestingly for speed, there were no significant differences between participant ratings ($\chi^2_3(N=26) = 7.20, p = 0.066$), which may have been due to the fact that the different selection scenarios all had different requirements, making certain techniques faster than others when coupled with errors. For accuracy, significant differences were found across the four techniques ($\chi^2_3(N=26) = 23.99, p < 0.0001$). As expected, study participants rated SQUAD ($Z = -3.36, p < 0.0083$), Expand ($Z = -3.25, p < 0.01$), and Zoom ($Z = -3.24, p < 0.0125$) all significantly higher than Raycasting.

Fig. 19 shows study participant’s most and least preferred technique. There were no significant differences in either the most preferred ($\chi^2_3(N=26) = 0.85, p = 0.84$) or least preferred rankings

($\chi^2_3(N=26) = 7.48, p = 0.058$). This result appears to stem from the fact that the techniques worked better or worse depending on the selection scenario.

5 DISCUSSION

Many of the outcomes were statistically significant which enable us to draw multiple meaningful conclusions. For [H1], the literature and our experience created an expectation that raycasting would be best suited for static, low-density environments. Scenario three showed that in a static low-density environment with minimal occlusion, raycasting was the fastest technique. However, for dynamic (scenario 1) and high object density (scenario 5), raycasting’s performance decreased as expected. In these cases, the other three techniques, designed for these cases, have an advantage.

For [H2], we expected that Zoom would be marginally better than raycasting for speed and accuracy. Analysing the mean completion time across all scenarios, you can see that Zoom is actually a little slower than raycasting, although the difference is not significant. When looking at total errors, Zoom is shown to have approximately half the errors of raycasting. With a similar speed and half the errors, Zoom has an advantage over raycasting.

For [H3], we proposed that raycasting and Zoom would not perform as well as SQUAD and Expand in dense, dynamic environments. In scenario two, Expand had significantly fewer errors and took significantly less time than both raycasting and Zoom. SQUAD however was not significantly faster, possibly due to the high number of iterative steps required for the large number of objects initially selected. SQUAD also performed poorly in scenario 4 due to the density of objects. SQUAD is somewhat better if accuracy is required, yet is hampered by object density.

For [H4], we proposed that Expand would be at least marginally faster than SQUAD in both static and dynamic scenarios. For the most part, this was true, with scenario 1 being the only exception. While SQUAD can handle faster motions found in scenario 1 and 2, SQUAD's issues with high object density keep it from performing better in scenario 2, where Expand is only moderately affected by object density. Based on these observations, we conclude that Expand is generally faster than SQUAD, and is significantly faster with higher object density. From these results, we have developed a set of preliminary guidelines:

- Raycasting remains a good general purpose selection technique under normal conditions.
- SQUAD remains accurate and fast in dynamic scenes, so long as the object density remains relatively low.
- The Expand technique adds a small amount of overhead to raycasting but performs better under difficult conditions.
- The Expand technique performs faster than SQUAD when object density is high.

These guidelines are in-line with the prevailing notion that there is no best technique for all situations. The best technique remains dependent upon the factors of the environment and there are many ways to tailor the technique to these needs.

6 FUTURE WORK

We have provided a foundation for exploring 3D selection in game-based VEs by examining four techniques under common object density/dynamics scenarios. However, one of the limitations of this work is that we did not examine the myriad of other 3D selection techniques found in the literature [6] nor did we fully explore all of the object density/dynamics combinations. Thus, with more effort, it may be possible to develop predictive computational models of how these and other selection techniques respond to dense and dynamic environments. This could enable automated methods to intelligently select techniques based upon the current environment. We plan to continue to explore this space and to review and revise existing 3D selection technique guidelines.

7 CONCLUSION

We have presented a summative evaluation of four different 3D selection techniques in five game-like VE scenarios that focus on high object density and dynamics. Raycasting and SQUAD are 3D selection techniques in the literature that are specifically designed for sparse and dense objects. We modified these techniques based on an iterative design process and developed two variations, Zoom and Expand. The results of our study indicate that the Expand technique performed significantly faster and with more accuracy over all the different scenarios. However, when examining the scenarios individually, each technique had better performance in terms of accuracy or speed depending on the level of object density and

movement. As such, we created four guidelines to assist designers with selecting between these variants and establish some preliminary guidelines for dense and dynamic 3D object selection.

8 ACKNOWLEDGEMENTS

This work is supported in part by NSF CAREER award IIS-0845921 and NSF awards IIS-0856045 and CCF-1012056. We would also like to thank the members of the ISUE lab for their support and the anonymous reviewers for their useful comments and feedback.

REFERENCES

- [1] F. Argelaguet and C. Andujar. Efficient 3D Pointing Selection in Cluttered Virtual Environments. *IEEE Computer Graphics and Applications*, Vol. 29, Issue 6, 2009.
- [2] D. Bowman, J. Chen, C. Wingrave, et. al. New Directions in 3D User Interfaces. *The Int. Journal of Virtual Reality*, 2006, 5(2), pp. 3-14.
- [3] D. Bowman, J. Gabbard, and D. Hix. A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. *Presence: Teleoperators and V.E.*, 2002, 11(4), 404 - 424.
- [4] D. Bowman and L. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *Symposium on Interactive 3D Graphics*, 1997, pp. 35-38
- [5] A. Forsberg, K. Herndon, and R. Zeleznik. 1996. Aperture based selection for immersive virtual environments. *ACM Symposium on User interface software and technology (UIST '96)*. ACM, New York, NY, USA, 95-96.
- [6] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004.
- [7] G. De Haan, M. Koutek, and F. Post. *IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection*. EGVE, 2005.
- [8] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 2, 65-70, 1979.
- [9] R. Kopper, F. Bacim, and D. Bowman. Rapid and accurate 3D selection by progressive refinement. *IEEE Symposium on 3D User Interfaces (3DUI)*, 2011.
- [10] J. Liang and M. Green. JDCAD: a highly interactive 3D modeling system. *Proceedings of Third International Conference on CAD and Computer Graphics*, 1994, pp.217-222.
- [11] J. Lucas, D. Bowman, J. Chen, and C. Wingrave. Design and Evaluation of 3D Multiple Object Selection Techniques. *Symposium of Interactive 3D Graphics and Games*, 2005.
- [12] A. Olwal and S. Feiner. The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality. *ACM Symposium on User interface software and technology (UIST 2003)*, 2003.
- [13] D. Rosa and H. Nagel. Selection Techniques for Dense and Occluded Virtual 3D Environments, supported by Depth Feedback. *XXIX International Conference of the Chilean Computer Science Society*, 2010.
- [14] A. Steed. Towards a General Model for Selection in Virtual Environments. *IEEE Symposium on 3D User Interfaces (3DUI)*, 2006.
- [15] Unity 3D Game Development Tool, available at <http://unity3d.com>
- [16] L. Vanacken, T. Grossman, and K. Coninx. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. *IEEE Symposium on 3D User Interfaces (3DUI)*, 2007.
- [17] C. Wingrave. 3DUI Flavors Beyond Vanilla. *SEARIS Workshop*, 2009.
- [18] C. Wingrave and D. Bowman. Baseline Factors for Raycasting Selection. *Proceedings of HCI International*, 2005.
- [19] C. Wingrave, D. Bowman, and N. Ramakrishnan. Towards preferences in virtual environment interfaces. In: *Proceedings of the Eurographics Workshop on Virtual Environments*, 2002.
- [20] C. Wingrave, R. Tintner, B. Walker, D. Bowman, L. Hodges. Exploring Individual Differences in Raybased Selection; Strategies and Traits. In *Proceedings of the 2005 IEEE Conference on Virtual Reality (VR '05)*.