2014

# Intelligent Selection Techniques For Virtual Environments

Jeffrey Cashion
*University of Central Florida*

University of
Central
Florida

STARS
Showcase of Text, Archives, Research & Scholarship

INTELLIGENT SELECTION TECHNIQUES
FOR VIRTUAL ENVIRONMENTS

by

JEFFREY ANDERSON CASHION
B.S. University of North Florida, 2009
M.S. University of Florida, 2010

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2014

Major Professor: Joseph J. LaViola Jr.

# ABSTRACT

Selection in 3D games and simulations is a well-studied problem. Many techniques have been created to address many of the typical scenarios a user could experience. For any single scenario with consistent conditions, there is likely a technique which is well suited. If there isn't, then there is an opportunity for one to be created to best suit the expected conditions of that new scenario. It is critical that the user be given an appropriate technique to interact with their environment. Without it, the entire experience is at risk of becoming burdensome and not enjoyable.

With all of the different possible scenarios, it can become problematic when two or more are part of the same program. If they are put closely together, or even intertwined, then the developer is often forced to pick a single technique that works so-so in both, but is likely not optimal for either, or maybe optimal in just one of them. In this case, the user is left to perform selections with a technique that is lacking in one way or another, which can increase errors and frustration.

In our research, we have outlined different selection scenarios, all of which were classified by their level of object density (number of objects in scene) and object velocity. We then performed an initial study on how it impacts performance of various selection techniques, including a new selection technique that we developed just for this test, called Expand. Our results showed, among other things, that a standard Raycast technique works well in slow moving and sparse environments, while revealing that our new Expand technique works well in denser environments.

With the results from our first study, we sought to develop something that would bridge the gap in performance between those selection techniques tested. Our idea was a framework that could harvest several different selection techniques and determine which was the most optimal at any time. Each selection technique would report how effective it was, given the provided scenario conditions. The framework was responsible for activating the appropriate selection technique when the user made a selection attempt. With this framework in hand, we performed two additional user studies to determine how effective it could be in actual use, and to identify its strengths and weaknesses. Each study compared several selection techniques individually against the framework which utilized them collectively, picking the most suitable. Again, the same scenarios from our first study were reused. From these studies, we gained a deeper understanding of the many challenges associated with automatic selection technique determination. The results from these two studies showed that transitioning between techniques was potentially viable, but rife with design challenges that made its optimization quite difficult.

In an effort to sidestep some of the issues surrounding the switching of discrete techniques, we sought to attack the problem from the other direction, and make a single technique act similarly to two techniques, adjusting dynamically to conditions. We performed a user study to analyze the performance of such a technique, with promising results. While the qualitative differences were small, the user feedback did indicate that users preferred this technique over the others, which were static in nature.

Finally, we sought to gain a deeper understanding of existing selection techniques that were dynamic in nature, and study how they were designed, and how they could be improved. We scrutinized the attributes of each technique that were already being adjusted dynamically or that could be adjusted and innovated new ways in which the technique could be improved upon. Within this analysis, we also gave thought to how each technique could be best integrated into the Auto-Select framework we proposed earlier. This overall analysis of the latest selection techniques left us with an array of new variants that warrant being created and tested against their existing versions.

Our overall research goal was to perform an analysis of selection techniques that intelligently adapt to their environment. We believe that we achieved this by performing several iterative development cycles, including user studies and ultimately leading to innovation in the field of selection. We conclude our research with yet more questions left to be answered. We intend to pursue further research regarding some of these questions, as time permits.

# ACKNOWLEDGMENTS

The completion of this dissertation would not have been possible without the support and guidance of my family and friends. I would like to specifically address and acknowledge the following groups and individuals for their contributions.

- My parents, for always believing in me and supporting me in all that I do.

- Dr. LaViola, my advisor, who has guided me through the years, both as a mentor and as a friend.

- Dr. Bassiouni, for reaching out to me and encouraging me to write my first paper. The continued collaboration served both of us well for many papers afterwards.

- Dr. Hughes, for pushing the limits of what I thought was understandable by humans, and helping me through the hardest course of my life.

- Dr. Bowman, for genuinely challenging my way of thinking about my research area, and providing valuable advice on how to improve my work.

- My friends who have stood by me and ensured that I put my education first, even when there was a mountain of business-related work to do otherwise

- Dr. Ash Patel and the "Euro Trash" gang, who included me into their group during the VR2014 conference and made me feel welcome.

- All of my lab mates in the ISUE lab at UCF; your support and companionship over the years has made my experience that much more enjoyable.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

Selection techniques are a common component of a Virtual Environment (VE) [8]. VEs can exist in many forms, ranging from a two-dimensional interface on a standard display to a three-dimensional world that is presented via a head-mounted display (HMD). In recent years, the range of environments where VEs exist has grown to include smaller devices such as tablets and smart phones [42]. No matter the form, the experience will involve the necessity that the user must often times select an object of importance. This may be a menu item, an action button, or even a 3D object, to name a few. The selection technique that is made available for the user to utilize to perform this task is therefore critical to not only the act of selection, but the success of the VE as a whole.

Much work has been done in the area of selection technique design, in a myriad of conditions [1] [10] [19]. This has caused the quantity of selection techniques available for use to grow to a sizable level, giving developers many options [2]. The most basic of these is a point cursor. It is a direct analogy to pointing on the screen to indicate what exactly is desired for selection. This extends into the 3D world by projecting a ray into the environment, but is still fundamentally the same. From this, more advanced techniques were developed that sought to address performance issues with the previous techniques [33] [20]. While many techniques are incremental in nature, others take a more dramatic shift, and change the way selection is approached. Such an example is the Ninja cursor, which actually uses multiple cursors at the same time, and allows the user to use the one closest to the desired object for selection [34]. More recently, several selection techniques have been designed that break down the

selection process into multiple steps. Both Expand and SQUAD operate in this manner, albeit in a different way. They achieve significantly better performance than a standard point cursor, and also work better across a larger range of conditions.

Typically, designers must decide on a single technique to implement, one that works best on average over the spectrum of expected conditions. This works fine in VEs where the expected conditions are relatively constant, but this is not necessarily the case when conditions vary dramatically [10]. Due to the ever increasing processing power available, better visuals and more complex scenarios are becoming much more common; it is becoming harder to accept a one size fits all mentality with respect to choosing which technique to use. It is our goal to provide a better understanding of selection techniques when faced with this challenge of complex scenarios, and see what progress can be made with the implementation of more novel dynamic selection techniques.

Efforts have already been made to include multiple techniques into a single framework that is responsible for choosing the most appropriate at any given time [25]. This type of framework seeks to address the challenges faced when a VE possesses very dynamic scenarios. It accomplishes its goal by analyzing the scene on a frame by frame basis, and determining which technique is best suited for those conditions measured for each frame. This work was theoretical in nature, not including any real selection studies. There is still a need to implement some of these ideas and put them through some real world testing, as we have done in our work.

## Section 1.1: Statement of Research Question

Scenarios that users are placed in vary greatly. The decision to use a particular selection technique must be made with the knowledge of where it will be used, and how well it is expected to perform. In recent years, challenges have been proposed that seek to inspire innovation in the field of selection techniques. The types of scenarios that a selection technique can be expected to operate in have virtually no limits. How, then, should a selection technique designer create their technique? There are standardized tests for selection, but these do not truly test what will be encountered in the real world. As is the case with a lot of things, synthetic testing does not always correlate to real world performance. The ideal strategy would be to have a selection technique that is prepared for anything, able to adapt to its environment on the fly.

If we are to truly study the performance of 3D selection techniques, then it would be beneficial to have a set of test conditions for which to test them in. These conditions will vary according to common and obvious features, ones that should apply regardless of other more specific factors.

Once we establish a baseline for testing and obtain initial results, we are still left with techniques that are revealed to work best on a subset of conditions, but never all conditions. We propose a framework for switching between techniques to offer the best experience, and also to act as a software platform for future testing. This framework can be modified by anyone, but our initial efforts will provide enough to get started.

The recent design of new selection techniques largely seem to be tested in 2D conditions, which is less than appropriate for determining their actual performance in 3D conditions. We would like to take these techniques and scrutinize them, identifying what parts of them are most critical to their performance, and come up with proposed solutions to address any shortcomings they may have. If we can perform a thorough study of them, we can also identify how they can best operate within our framework, which would improve the value of the framework and likelihood of future use.

In short, we propose the following list of research questions:

- How do we identify which environment(s) a selection technique is better suited for?
- What would a more relevant form of testing look like if it was designed to compare the relative performance of existing and future techniques?
- How can we engineer a better way of matching techniques to their optimal environment(s)?
- In what ways can we improve techniques such that their performance envelope is extended to include more flexibility?

Section 1.2: Scope of the Research

As with many fields, selection theory is quite broad. It would be impractical to not focus on a narrow area, so we will only cover a smaller domain of selection. For the majority of our work, the display interfaces used are standard 2D screens, such as televisions or

computer monitors. We exclude 3D headsets and augmented reality headsets, as well as displays that simulate 3D using some sort of special glasses. While these could benefit from our work, they are not explicitly tested.

For input devices, our work applies most to common devices such as computer mice, Sony Move controllers, and possibly the Microsoft Kinect. We are not focused on the design or efficacy of these devices; we are only interested in them so far as they are required to attain the proper input methods that we need to have for our techniques that we study.

With regards to selection itself, we are most interested in the accuracy of selection, but also take a strong interest in which technique performs the quickest. In some cases, one of these would be more important than the other, and when relevant, it will be noted as such. The selection techniques that we are most interested in focusing on are those that act in a 2D plane on top of a 3D environment, but may also exist purely in 2D. Many times, a technique that works well in 3D can be simplified to work in 2D by discarding depth data, but this is not always the case.

Section 1.3: Contributions

Contextual analysis of a scenario for enhanced selection purposes is a relatively unexplored area of research. We have created a new selection technique designed to work well in dense and dynamic environments. Within the same user study, we also determined the performance characteristics of three other selection techniques, and

made some general guidelines for 3D selection in games and simulations. We have also created the initial platform for establishing a more robust optimal selection technique determination algorithm with our Auto-Select Framework (ASF). Our framework shows that there is potential for choosing between techniques on the fly, but also highlights many issues that must be further explored.

In response to our ASF research, we investigated the design flow of creating a dynamic technique by joining to existing selection techniques, and identifying how the selection technique should dynamically adapt to its environment. The primary driver for this design method was to avoid some of the negative aspects of the ASF. Following the design and testing of this technique, we studied several existing dynamic selection techniques, identifying potential modifications that could lead to improved performance, as well as how they might integrate well into our ASF.

In summary, the following contributions are:

- A performance comparison of selection techniques across different environments, which served as a foundation for future research.
- The design of multiple new selection techniques, some of which adapt to their environment dynamically.
- The creation of a framework that automatically selects between different selection techniques in real-time to improve performance. This allowed us to collect user response and the impact that various aspects had on performance.
- Develop a means of identifying selection techniques which can be combined to form one hybrid technique, with the goal of improving the selection experience

- A deeper understanding of the design and implementation of modern dynamic selection techniques, which lead to many proposed improvements left for future work.

- An analysis of modern dynamic selection techniques and how they can be incorporated into our auto-select framework.

## Section 1.4: Organization

In Chapter 2, we discuss in great detail the current state of selection techniques, and outline their defining characteristics. The techniques will also be given classifications to place them in a logical relationship with each other. Additionally, recent work on context-aware selection technique assignment is reviewed. In Chapter 3, we discuss the various taxonomies of selection scenarios, including their specific demands they place on the user. This will include our user study performed which specifically studied how different selection techniques operated in various dense and dynamic scenarios. In Chapter 4, we discuss the intricacies of optimal selection technique assignment, as well as discuss results found during our user studies which focused on exploring a framework designed to perform this optimal assignment. Chapter 5 discusses our development of a dynamic selection technique that was created by joining two existing selection techniques together. Chapter 6 discusses existing dynamic selection techniques and how they can be both improved and integrated into our Auto-Select Framework. Chapter 7 features a discussion of various topics. Finally, Chapter 8 has our conclusions and future work.

# CHAPTER 2: RELATED WORK

## Section 2.1: Selection Techniques

Many selection techniques have been designed throughout the years. One thing they all have in common is the principle idea of a user pointing to an object on the screen, and selecting it. We have classified existing techniques based on how they operate and if they are designed for 2D or 3D environments.

### Section 2.1.1: Static 2D Techniques

A static technique is one that does not change any of its parameters during runtime. The appearance remains the same to the user, with only slight visual feedback, if any, which might be used to inform the user that an object is within the selection area. Although static at runtime, it can be adjusted and tuned by the designer beforehand to best suit the expected conditions.

The standard point cursor is the most basic of selection techniques [8]. It involves pointing a cursor, usually an arrow or cross-hairs, at a target. Selection is performed in one step, and occurs at a single point on the cursor. No contextual information is utilized; only the x and y positions of the cursor are used to determine where a selection is made. This is very common in 2D interfaces, but is also applied to a 2D plane (such as the display surface) in a 3D interface. This technique was tested by Fitts [22] during his initial research when modeling the index of difficulty of selection. From this

elementary technique, others are constructed with increased intelligence and complexity.

The Prince Technique [33], named after the tennis racket manufacturer, takes the basic point cursor technique and changes the selection point to a selection area, such as a rectangle or circle. This makes selecting a target that is very small much easier, at the cost of possible ambiguity. With a single point of selection, it is next to impossible to have an ambiguous selection choice, as the single pixel can only overlap one object. With the area cursor in the prince technique, multiple objects can be within the selection area of the cursor, thus leading to ambiguity. This technique was compared directly to the technique used in the original Fitts' law research, which involved moving a cursor sideways to select an object. The performance of this technique strongly agrees with the predicted performance modeled by Fitts' law. This technique only applies to 2D selection.

*Section 2.1.2: Dynamic 2D Techniques*

Based on the area cursor, the Bubble Cursor [26] seeks to address the issue of target ambiguity. Instead of having a static area that could contain multiple targets, the bubble cursor dynamically adjusts the perimeter of the area of the selection circle so as to only encompass a single target. The initial cursor exists as a circle. If it is possible to include the nearest target by growing the radius without including the second-nearest target, then the radius is increased. If doing this would cause the inclusion of a second target, then the radius is increased as much as possible before including the second object,

then the shape of the area is selectively modified to include only the nearest target, thus avoiding ambiguity. The remainder of the bubble remains the same, and the center of the bubble remains in the same position as it would if no adjustments to its shape were made. It is worthy to mention that the effective areas that relate to which target the cursor associates with is equivalent to the Voronoi diagram. This technique only applies to 2D space.
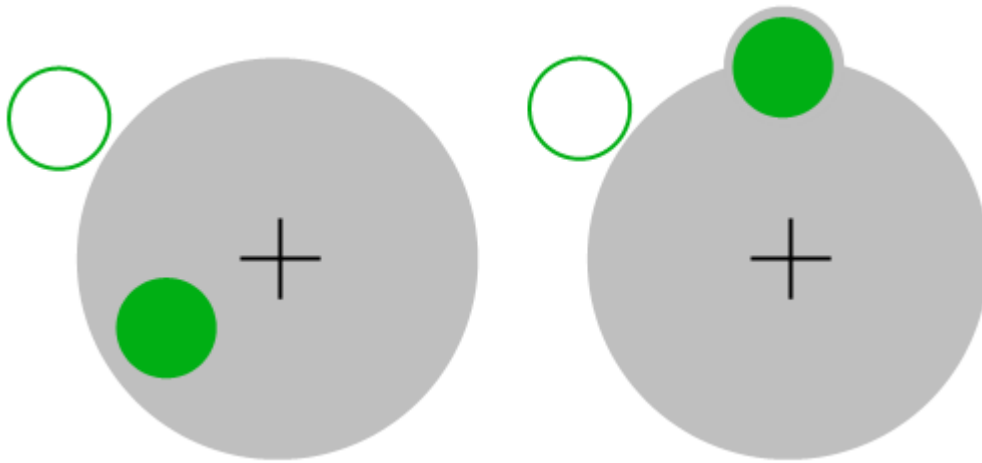


Figure 2.1: Bubble Cursor grows to include closest target (left). Bubble is grown to only overlap nearest target, then morphs to include it fully (right).

DynaSpot [14] seeks to address the issue of target ambiguity that plagues the area selection technique. This is done by dynamically varying the radius of the selection circle as a function of cursor speed. As the users cursor speed increases, so does the radius of the circular area cursor. This makes tracking and selecting smaller and faster moving objects easier. When stationary, the selection area acts more like a point cursor, allowing precise selection and reduced ambiguity. When moving, it allows for easier selection of moving targets than point cursor. The cursor size is not directly linked to

cursor speed, but instead is loosely based on it, which is the product of an algorithm that allows for ramp-up and ramp-down of the size gradually instead of a tight 1:1 response to user input. These values were derived during pilot testing of the technique.

The Ninja cursor seeks to reduce selection time by actually having multiple cursors on the screen at any one time [34]. The way this is accomplished is by reducing the effective distance from any one of the cursors to the user's desired target. All cursors move in the same direction and at the same velocity. If multiple cursors have an object beneath them, then the one that is closest to the center of the object is the one that becomes active, while the others lay dormant. The others are actually nudged out of the way so as to not overlap with any objects. This selection technique only applies to 2D environments. Further improvements have been developed which help to determine which cursor the user is focused on by leveraging eye-gaze [50]. This type of technique, which differs quite strongly from traditional single-cursor selection techniques, is not without its own design concerns, and should be used with care [49].

*Section 2.1.3: Static 3D Techniques*

Raycasting, also known as laser pointing, is the 3D version of point cursor [38]. It involves a ray being projected into the 3D world. It is generally the baseline to which other 3D selection techniques are compared. Often, the ray is projected strait into the scene, and acts as if it was a point cursor acting on the front plane of a 3D world. More complex versions will provide the ability to rotate or angle the cursor as it enters the scene, thus allowing more complex selection [2]. This is often done with 6-DOF input

systems where the user has the ability to rotate and manipulate the selection input device. Rotating the raycast can allow the user to essentially reach around obstructions to reach their desired target, but is still prone to jitter and inaccuracy when selecting small or moving targets.

In much the same way that the Prince technique's area cursor improved over the point cursor, Spotlight does so over Raycast [38]. Instead of simply pointing a single ray with which to select, a cone, or spotlight is projected. It can either be directly projected with no rotation, or can allow rotation via user input, just like Raycast. The primary benefit of Spotlight is that it is easier to select small and or moving targets. The user is no longer required to be as precise as with Raycast. The primary downside to this technique is the problem of ambiguity, as with the Prince technique. To overcome this, developers must come up with their own solution, such as defaulting to the target closest to the center of the spotlight, shrinking the size of the spotlight, or some other higher-level method. The problem of selection ambiguity is one of the primary issues addressed in future techniques that are based on spotlight.

*Section 2.1.4: Dynamic 3D Techniques*

Aperture is based on the Spotlight technique, with the added advantage that the user can adjust the size of the selection cone [24]. This is explained metaphorically as adjusting the aperture, since the user is moving a real object towards and away from their dominant eye to adjustment the size of the selection cone. By giving the user this control, they are able to more easily control the precision of their selection, reducing

ambiguity when needed. The primary advantage of this technique is that it closely resembles how we point to something in the real world. The biggest drawback with this technique stems from parallax problems where the user is focusing on the screen past the tracking device and seeing two of it, one with each eye. This can cause some disorientation and confusion, and is suggested to be alleviated by closing one eye, as one would when aiming a weapon or other projectile. There is also the possibility of the input device occluding part of the display.

IntenSelect takes Spotlight a step further by accumulating a score over time for objects that lie within the cone region, and then indicating which object has the highest score [20]. Only objects that lie within the conic region accumulate a score, and once they leave, their score rapidly decays. The rate of accumulation and decay are precisely tuned by several variables. The object that is predicted to be desired by the user is known as the intended object. If there is no such object, then none are indicated by feedback on the screen. Their results showed a slight advantage over regular Spotlight, but more testing is needed.

The Hook technique was designed to address selection in dense environments with moving objects [45]. It uses a heuristic to determine which target the user is attempting to track. This is done by measuring the distance from the cursor to each object, and assigning it a score, where a higher score is more favorable. For each time step, the list of object distances is sorted. Nearer objects are given a higher score than farther objects, and this scoring goes up or down, depending on where the object ranks in the list of all objects. This score accumulates over time, but also drops off as it gets older,

13

thus placing more emphasis on more recent samples. Overall, this causes the object that is typically closest to the cursor in the past moment to be hooked by the cursor. This is essentially Bubble Cursor, in 3D with some memory. The object with the highest score for each frame is determined to be the desired object. This is indicated to the user via a different visual style applied to it, such as a red tint versus grey for all the other objects. As with IntenSelect, the use of score accumulation over time can lead to an undesirable object being indicated as the target, for a brief moment while the scoring shifts in favor of a new object that the user is attempting to indicate. This effect can be minimized to some degree by modifying the scoring algorithm, but this can degrade the performance of the technique in other situations.

SQUAD was designed to address the problem of selection in dense environments [35]. It uses a spotlight area cursor to initially choose objects, then when the user selects, a second stage of selection is done. All objects are divided up into 4 quadrants on the screen, and the user then selects the quadrant that contains the initial object of interest. Objects not in that quadrant are discarded, while those in the quadrant are redistributed to the four quadrants, and the user repeats this process of iterative reduction until there is only the one desired object in a quadrant, which the user will presumably select and end the selection task. While effective, it can take many steps to achieve a single effective selection. For example, If 64 objects are initially selected, then three iterative reduction steps are required. This will result in a total of four user inputs to select a desired object from start to finish. This method is also taking the user out of context by breaking away from the original scene to do this iterative process. If environmental

context was needed, then this technique will fall short. Additionally, if there are multiple instances of the same object, but one is of more interest over another, then there is no way to know which one was the originally desired one, thus leading to a very frustrating selection experience. For this technique to be optimally utilized, it is best suited for static environments with no motion, where no context is required and objects that are identical to each other are also equivalent when selected.

Starfish is yet another technique designed to ease selection in dense environments [65]. The user is responsible for positioning a multi-legged starfish somewhere within the 3D scene. Each leg represents a connection from the head of starfish to a potential target. Once the user locks the starfish in position, the user traverses down one of the legs and selects the desired target. These two modes of manipulation are known as move and select, respectively. The potential targets are chosen using the distance, angle, and quantity of the objects to best determine those that are most likely desired by the user.

While not actually a selection technique, Lank et al. developed a way of predicting the endpoint of a cursor movement based on the motion kinematics [36]. Their methods resulted in an endpoint prediction accuracy of over 42%. If combined with other selection technique methods, it could be a powerful tool to increase the accuracy beyond what a technique could produce on its own.

*Section 2.1.5: Interactive 3D Techniques*

Blurring the line between static and dynamic is the Virtual Hand [30]. Existing in the scene is a virtual hand that mimics the position and orientation of the users input device. The input device can be a controller, a glove, or some other device. Either one or two hands can be used at the same time, allowing for a more immersive experience. Research has shown that using two hands improves efficiency and makes special input more comprehensible. The relative positions of the hands in the scene gives the user a reference point to gauge size and distance, since they can compare it to the distance of their real hands.

To overcome the problem of limited reach with the virtual hand, the Go-Go technique was created [47]. It works by increasing the effective reach of the hand after a certain point. From reach distance A to B, the real world to virtual world mapping of position is 1:1 or linear. Beyond point B, the reach can become super linear, such as exponential. For example, let's measure the distance from the user's chest to their hand. From a distance of zero to one foot towards away, the virtual hand moves a virtual foot. After one foot, the extra distance is squared to compute the virtual distance, so that at 3 feet, the virtual hand is now 5 feet away from the virtual chest. This gives the user an extra 2 feet of virtual reach that would not have otherwise been possible. This technique is great for reaching objects at a distance, and reduces the need to navigate. Also, it does not affect close-range reach, as that is still done in a natural 1:1 or linear mapping.

While the Go-Go technique is great for increasing reachable range, it does little for avoiding obstacles. The Flexible Pointer is a two-handed technique that allows the user to reach out with one hand and bend the tip of the cursor with the other [44]. It is possible to incorporate the Go-Go method into the reach, thus gaining its advantages. The second hand is used to steer or turn the end of the cursor, thus allowing the user to point past an obstruction, and then bend around to point at the desired object which may not have had a direct line-of-sight selection path. This can be especially true when a virtual scene uses collision detection to prevent the user's virtual hand or cursor from passing through objects. With the flexible pointer, the user can simply go around the obstruction to reach the desired object without having to navigate.

Within a head-tracked virtual environment, several new types of 3D interaction techniques are possible. Selection and manipulation metaphors which join the user's physical actions with the virtual environment should appear natural and seamless to the user [23]. One such technique is Head Crusher [46], which involves a user extending out their arm and hand, then use a pinching motion to pinch a virtual object between their fingers as if they could literally touch it. Another strongly metaphorical selection technique is the Sticky Finger, which requires that a user extend a finger over a desired object, which then permits them to interact with it. A third selection technique is offered, called the Framing Hands Technique. Similarly to how an individual might extend their hands out and box in an object between their hands using the index finger and thumb at a right angle, this technique requires the same action to indicate a desired object for further manipulation.

## Section 2.2: Context-Aware Selection Technique Assignment

The idea of context-sensitive selection is a relatively new one. Frees developed a context-driven interaction model, used for designing interfaces that rely on contextual information to work optimally [25]. This research was more of a theoretical analysis, and didn't include any user testing. He also created an initial software implementation to serve as a demonstration of how one could be created. His toolkit was also designed to be used by others as an example of how to use this context-driven model for enhanced selection performance. Octavia [43] looked in another direction for gathering contextual information; towards the user. By collecting information about them, the user interaction could be custom tailored to each user. They found that by adapting the characteristics of the techniques based on user input, the performance significantly improved and frustration decreased. This sort of adaptive technique refinement could be used in conjunction with information from the scene to create an even better performing selection technique or group of techniques.

## Section 2.3: Selection Enhancements and Tools

The need to select small targets has been addressed recently, including work on adapting the size of the objects or by providing other visual aid to the user [16] [17]. This method of visual expansion of potential targets can be easily seen in the consumer oriented feature of the OS X software icon tray, which enlarges the size of icons as the user is in close proximity to each one. In some cases, depending on implementation, a

fish-eye effect might be implemented. To aid in a reduction in visual distortion, research was done on minimizing this distortion, called speed-coupled flattening [28].

The ability to discriminate small objects in 3D environments was something which was improved by using pop-up depth cues [60]. This featured a set of additional 3D views that would appear when the user needed to visually separate one object from another in a confined 3D space.

# CHAPTER 3: DENSE AND DYNAMIC 3D SELECTION

## Section 3.1: 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 [8]. Current video games have detailed, interactive scenes created with advanced modeling 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" [61], we look for issues performing selections and find solutions to them through iterative design. We started with Raycasting [8] and SQUAD [35]. 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 [38] 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 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 modeled 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. 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.

## Section 3.2: Selection Techniques Studied

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.

### Section 3.2.1: Raycast

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.

*Section 3.2.2: SQUAD: Progressive Refinement*

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 (Figure 3.1). 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.
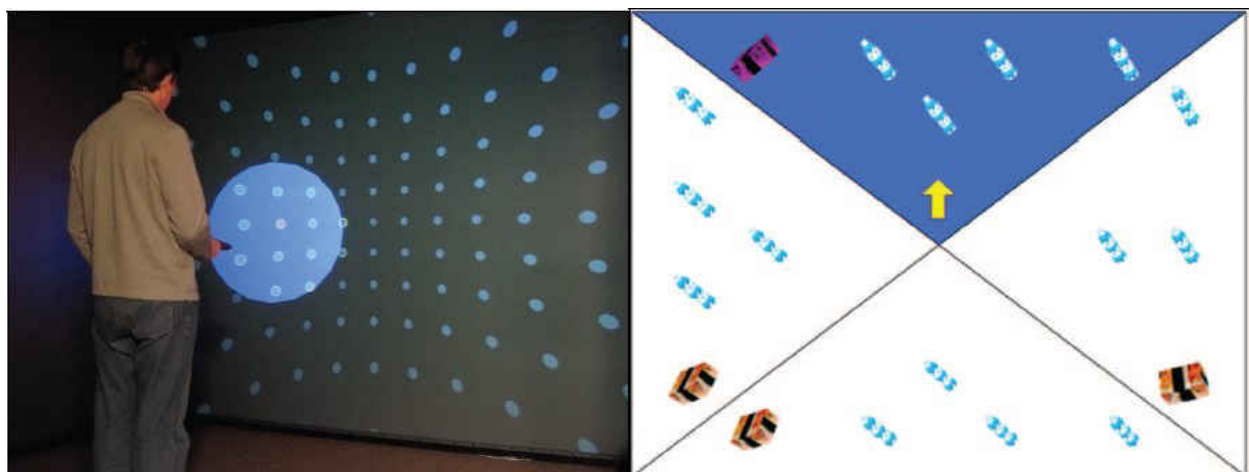


Figure 3.1: A user performs selection (left) and is presented with SQUAD (right)

23

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 techniques 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.

*Section 3.2.3: Zoom: Come closer, my pretty*

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 (Figure 3.2). 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.
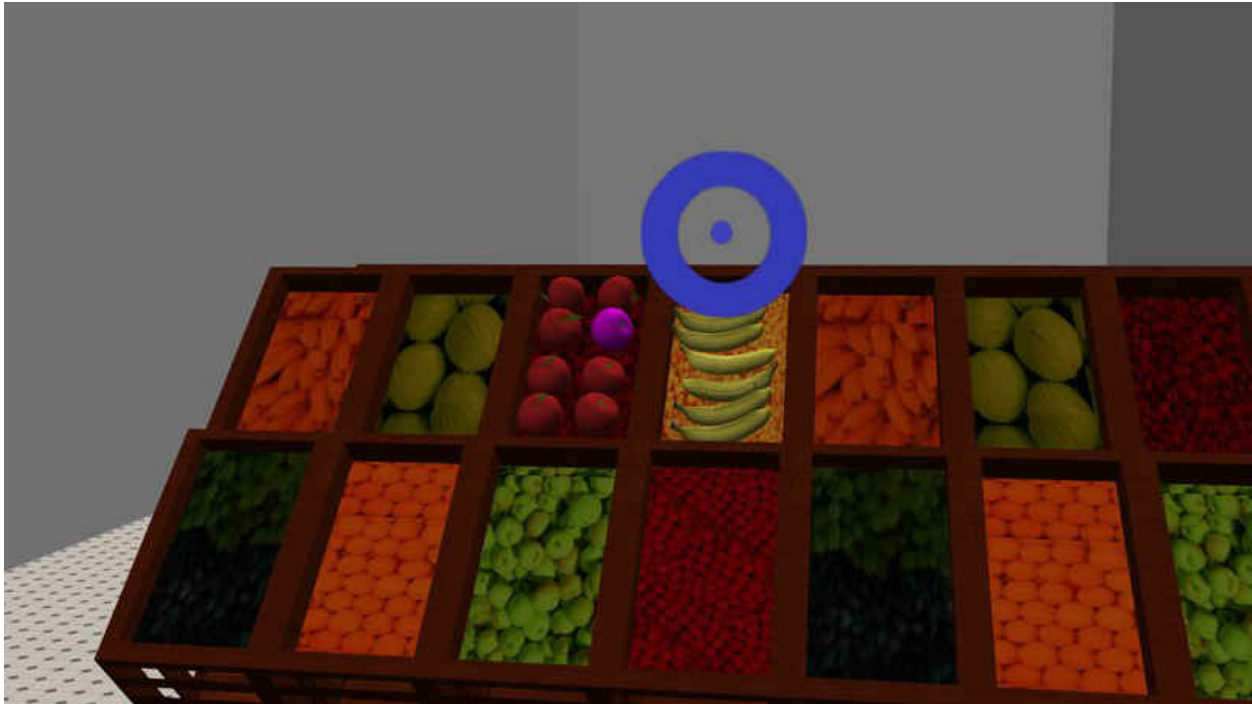


Figure 3.2: A user's perspective is zoomed in via the Zoom technique

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 the user zooms 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.

*Section 3.2.4: Expand: A Novel Selection Technique*

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 section 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.
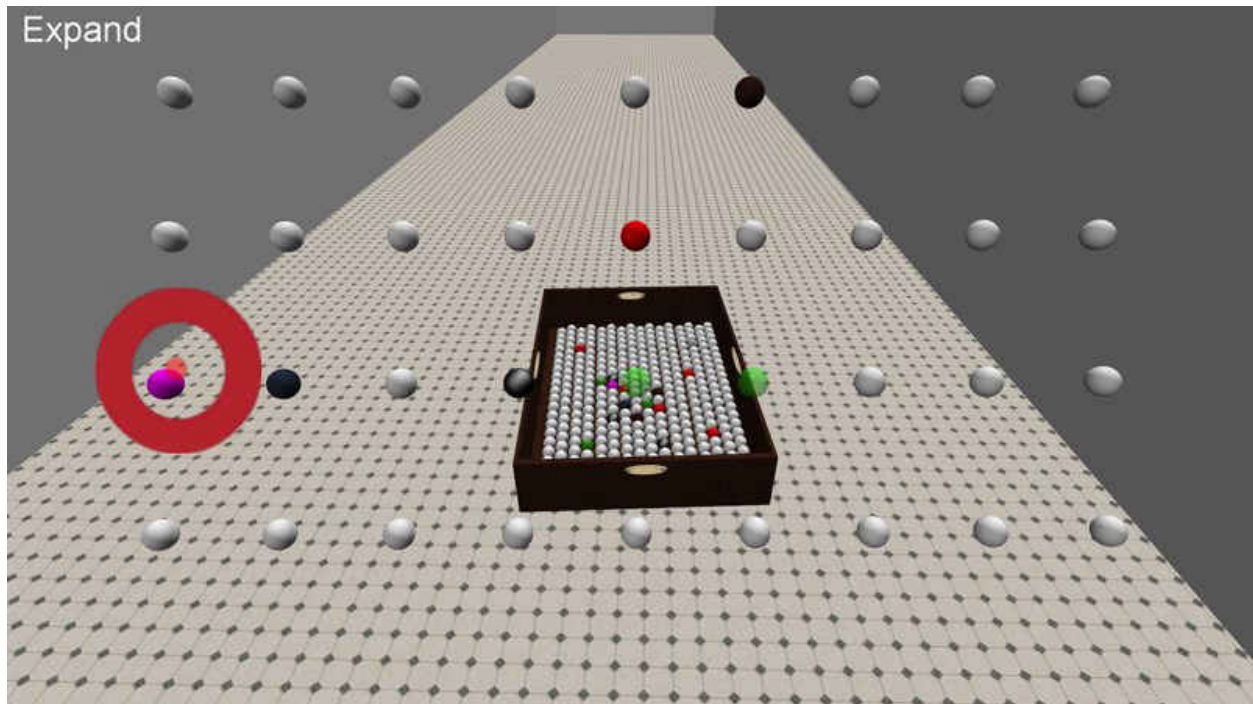
Figure 3.3: A user's perspective after making an initial Expand selection.

The quadrant arrangement used in SQUAD was modified to be a virtual grid that filled

the screen (Figure 3.3). 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.

## Section 3.3: 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.

*Section 3.3.1: 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 [57]. 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.

*Section 3.3.2: 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].

*Section 3.3.3: 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 were then presented with a pre-questionnaire (see APPENDIX A). 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 APPENDIX B). 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.

*Section 3.3.4: 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. In each scenario, the user is presented with a myriad of objects, with only a single one being the target object. The target object is indicated by a unique pink color.

Participants are presented with a large enclosed area which featured 40 floating cubes that move in a random manner with periodically changing directions (Figure 3.4). 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.
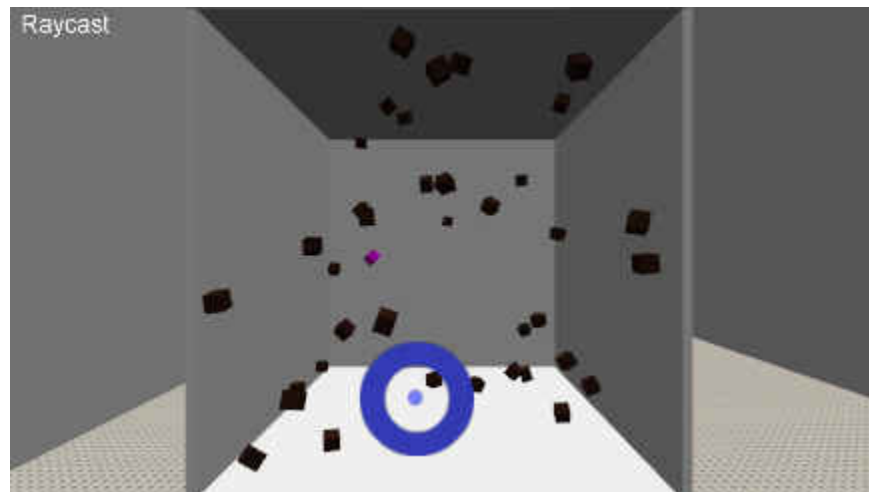


Figure 3.4: Scenario 1. The user is presented with a box that contains many cubes which are moving in an unpredictable manner

*Section 3.3.4.2: Scenario 2: High Density, High Motion*

Participants are presented with a rectangular tray which featured 320 small spheres of varying color (Figure 3.5). 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. The small size of the object, in combination with the speed of rotation made it very difficult to select the garget. This scenario required very fine motor skills and accurate object tacking.
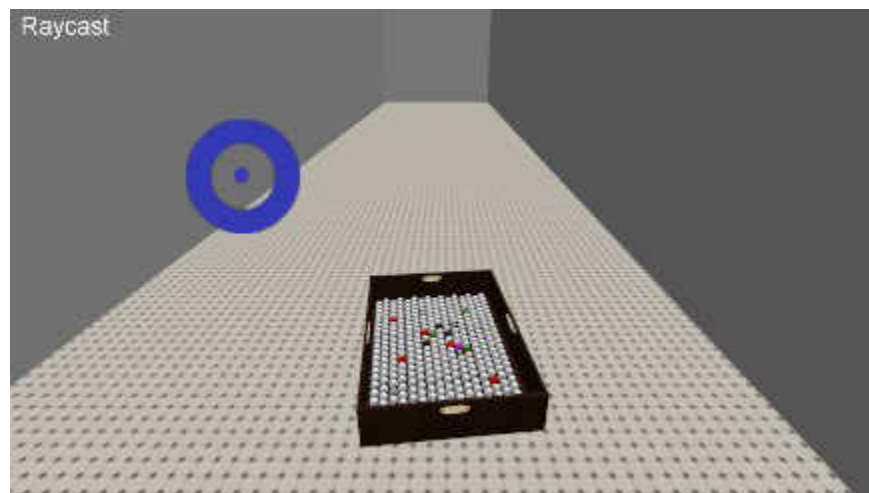


Figure 3.5: Scenario 2. The user is presented with a rotating tray, filled with 320 small round balls. Color is used to heighten difficulty.

*Section 3.3.4.3: Scenario 3: Low Density, No Motion*

Participants are presented with a fruit stand that featured several apples and bananas (Figure 3.6). 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.



Figure 3.6: Scenario 3. A fruit stand, which contains many sections filled only with a texture, but others with real 3D objects to select.

*Section 3.3.4.4: Scenario 4: High Density, Low Motion*

Participants are presented with a rotating table that features 42 medium-sized boxes of varying color (Figure 3.7). 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. The nature of the rotation and object placement caused the target to be blocked at certain times, thus requiring the user to strategically time their selection to match the opportunities when the objects was visible.
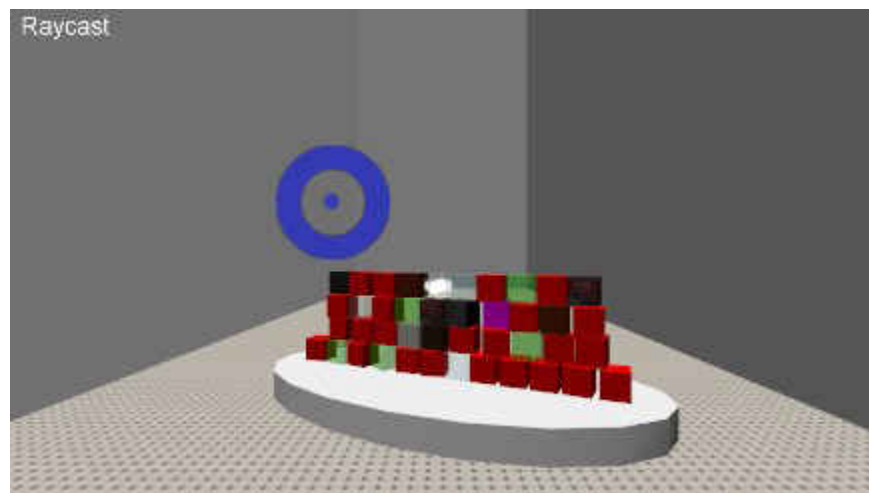


Figure 3.7: Scenario 4. A rotating table with 42 boxes of varying color. The target object is off-center, thus requiring object tracking.

Participants are presented with a table top that featured sixty-six medium-sized cans (Figure 3.8). 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.
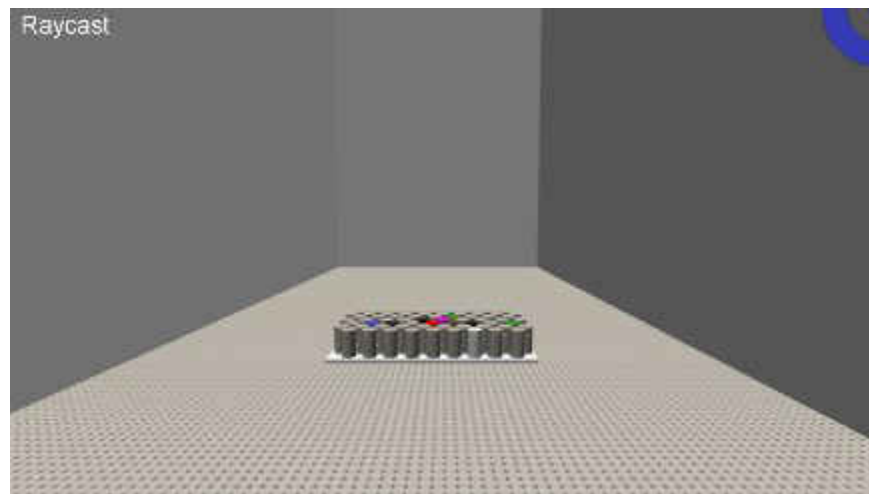


Figure 3.8: Scenario 5. A tabletop featuring 66 cans. There is a lot of occlusion which obscures the target object, increasing difficulty.

*Section 3.3.5: 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.

*Section 3.3.5.1: Overall*

Figure 3.9 and Figure 3.10 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 ($t_{25} = -4.96$, $p < 0.01$).

38
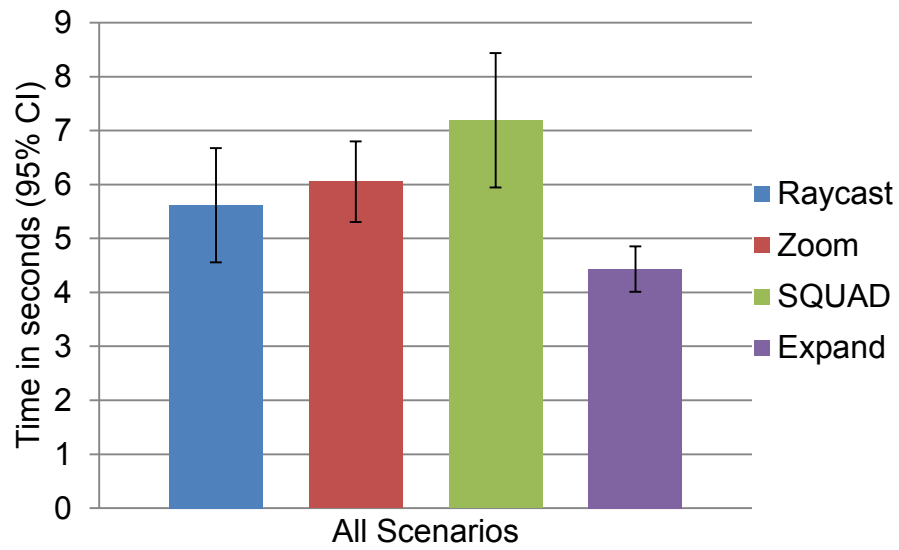
Figure 3.9: Mean Total Time, All Scenarios. Expand is significantly faster than Zoom

and SQUAD.
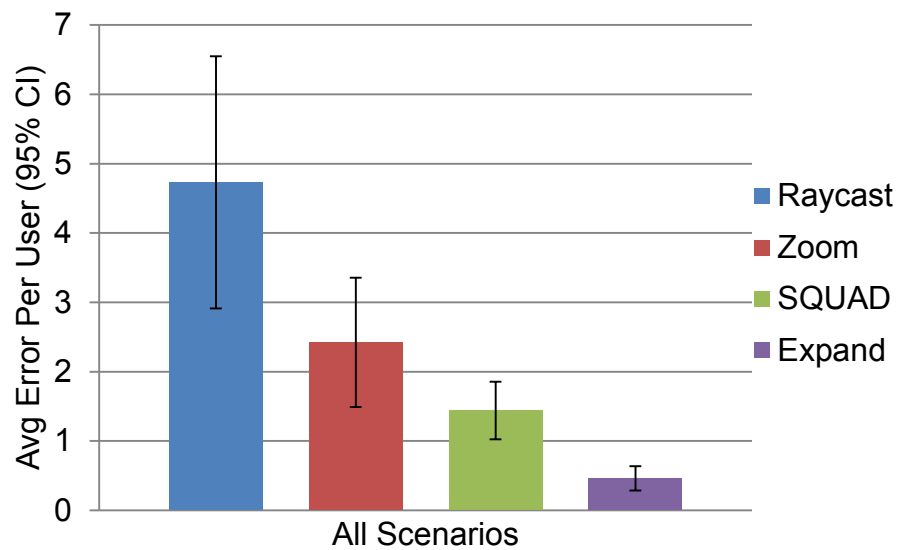


Figure 3.10: Average Errors, All Scenarios. Raycasting experienced the most errors,

followed by Zoom, SQUAD, then Expand.
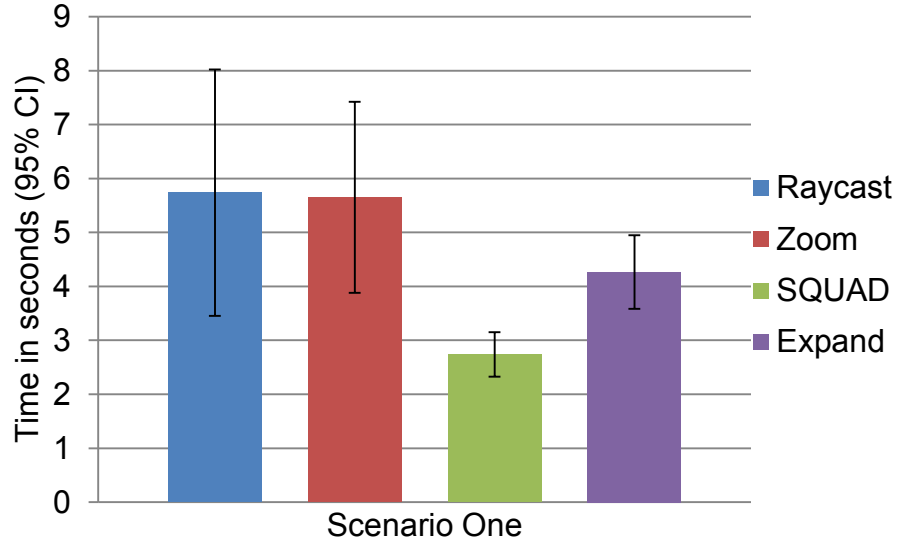
Figure 3.11: Mean Completion Time, Scenario 1. SQUAD experienced the fewest

number of errors, followed by Expand.
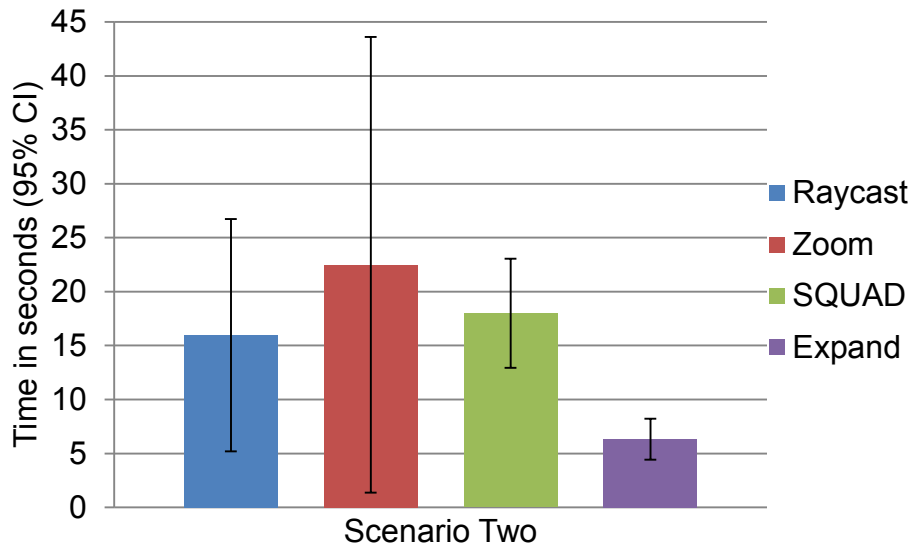


Figure 3.12: Mean Completion Time, Scenario 2. Expand was significantly faster than
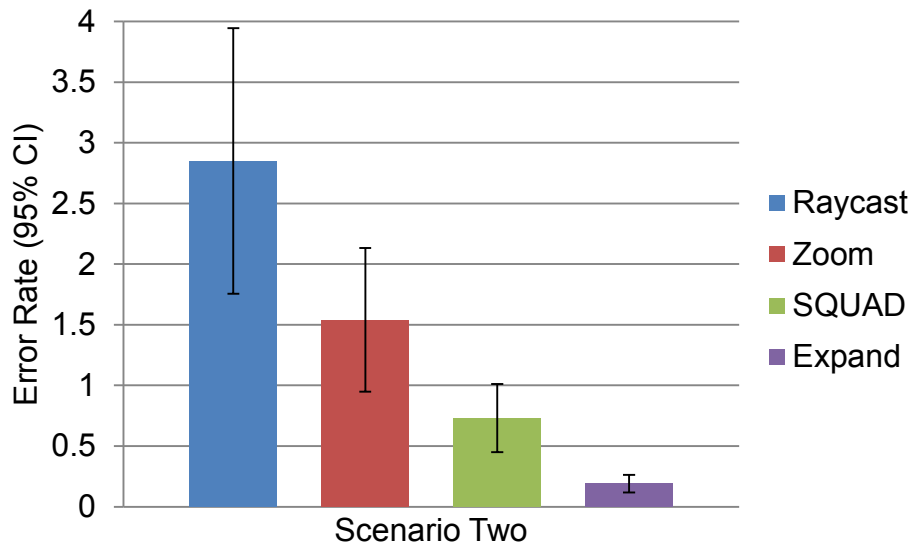
SQUAD.

Figure 3.13: Mean Errors, Scenario 2. SQUAD and Expand have significantly fewer errors than Raycasting, With Expand having significantly fewer errors than SQUAD.
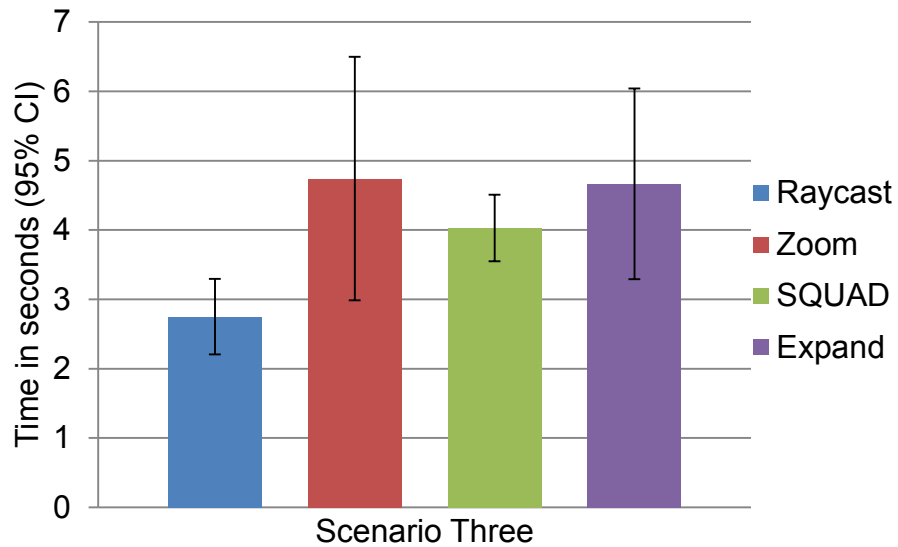


Figure 3.14: Mean Completion Time, Scenario 3. Raycasting was significantly faster than SQUAD due to simple and direct selection.
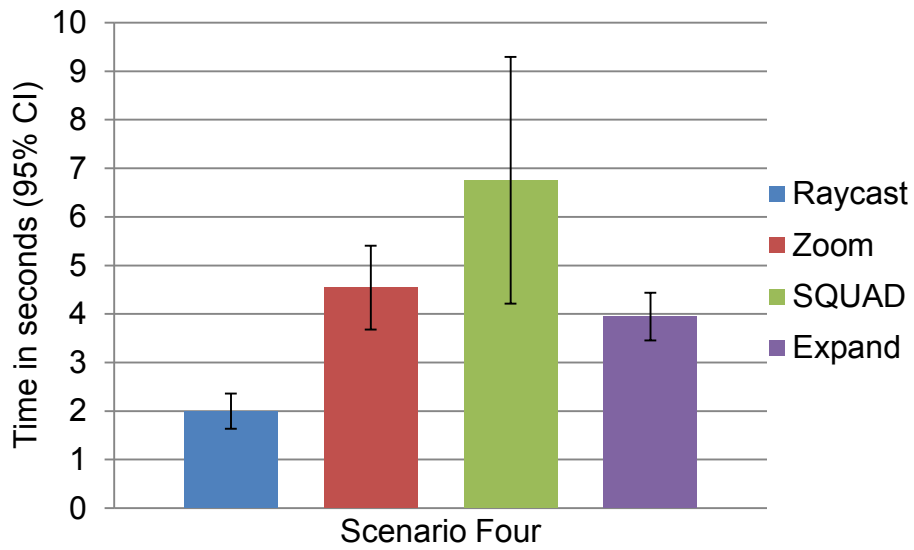
Figure 3.15: Mean Completion Time, Scenario 4. Raycasting is significantly faster than all other techniques.
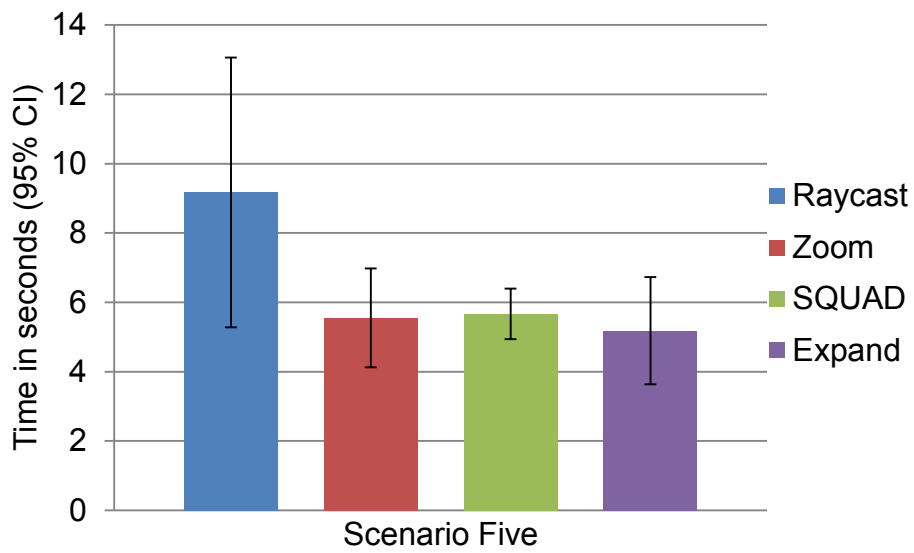


Figure 3.16: Mean Completion Time, Scenario 5. Raycasting was slower than other three techniques, but significance was weak.
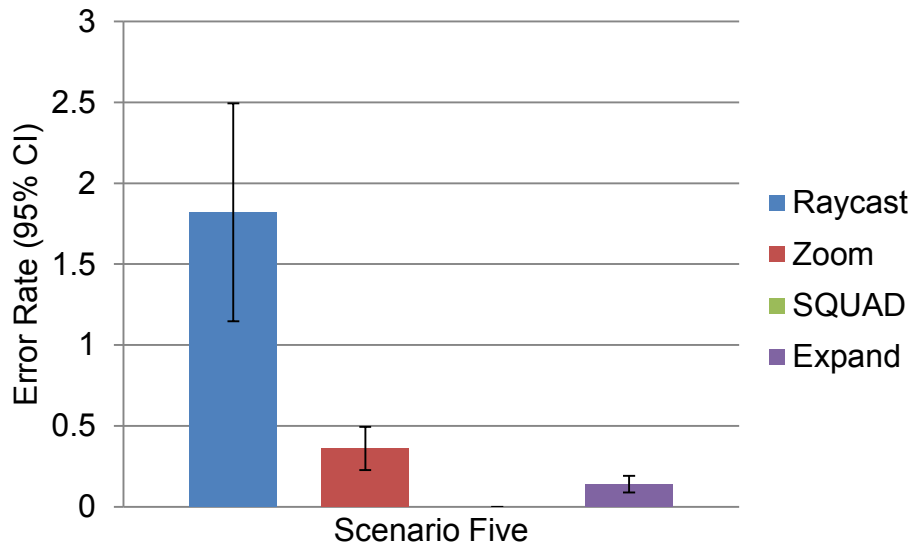
Figure 3.17: Mean Errors, Scenario 5. Raycasting had significantly more errors than the other techniques. SQUAD experienced no errors.

*Section 3.3.5.2: 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 (Figure 3.11), 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.

*Section 3.3.5.3: Scenario 2: High Density, High Motion*

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. Figure 3.12 and Figure 3.13 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.

*Section 3.3.5.4: Scenario 3: Low Density, No Motion*

Scenario 3 featured stationary objects that were not very dense. Figure 3.14 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 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 multistep 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.

*Section 3.3.5.5: 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. Figure 3.15 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, $p = 0.231$).

*Section 3.3.5.6: Scenario 5: High Density, No Motion*

Scenario 5 featured a table with cans situated where the target object was mostly occluded from all sides. Figure 3.16 and Figure 3.17 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 reattempting their selection when using Raycasting to make up for the difficulty associated with the low target object exposure.

*Section 3.3.5.7: 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 [31] was used at α = 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. Figure 3.18 shows the mean ratings for Q1-Q4. For usability, significant differences were found across the four techniques ($\chi_3^2$ (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.

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.

Figure 3.19 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.
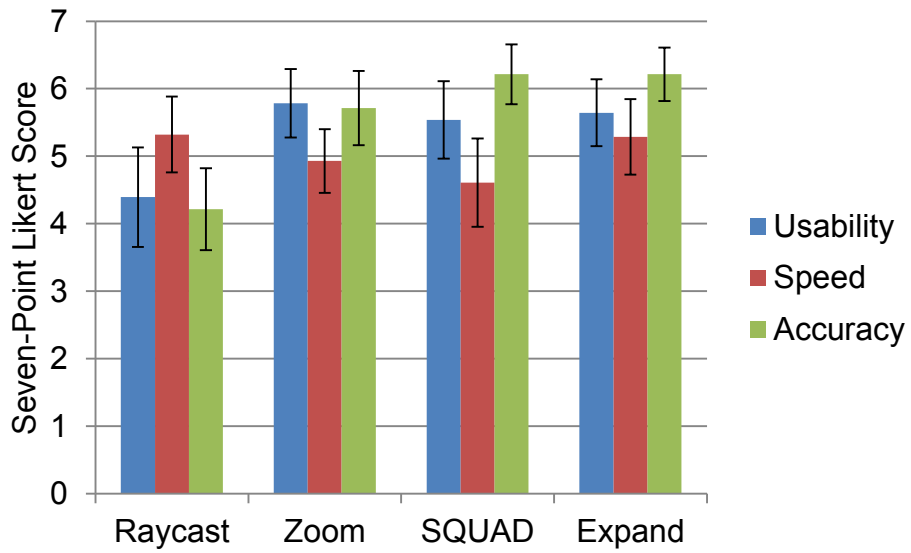
Figure 3.18: Post-Questionnaire, Technique Critique. Raycasting was less usable and less accurate than other three techniques.
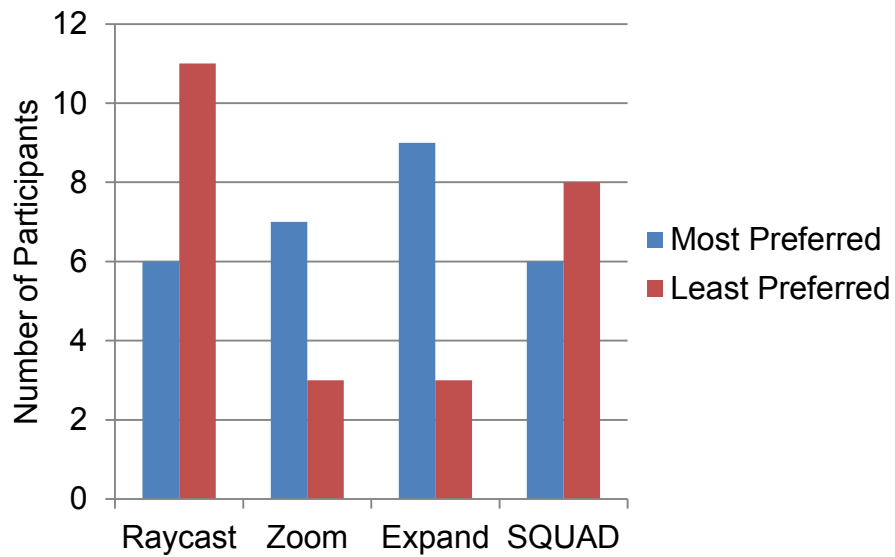


Figure 3.19: Overall Ranking of Selection Technique. There was no significant favorite among all techniques.

48

Section 3.4: 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), the performance of Raycast 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. Analyzing the mean completion time across all scenarios, it can be seen 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.

A broader implication of this research is that of the test environment, and how it relates to the current form of selection technique testing. Up until now, the majority of techniques have been tested using some form of ISO standard or some other trivial

environment, neither of which relate well to their performance in a more realistic and practical scenario [53]. In our case, we sought out to specifically design environments to go along with the techniques, so that they may be tested in more realistic scenarios. In addition to that, we also would like to put emphasis on the fact that we tested across a broad variety of scenarios, not just a single one. It is much more typical that techniques are only tested in a single scenario, regardless of how realistic it is. These two factors, both relativeness and quantity, provide a better way forward in testing selection techniques and obtaining performance metrics that are more reliable and relevant. In addition, we can leverage these reliable values to better determine which environment a selection technique is better [or best] suited for, which answers our initial research question.

# CHAPTER 4: AUTOMATIC TECHNIQUE ASSIGNMENT

## Section 4.1: Introduction

After completion of our initial study, we took a look at the performance of the techniques tested and the guidelines that we established. From this, we established the idea of transitioning between the techniques somehow, without distracting the user. If we could get raycast to operate when it was best suited, and likewise get Expand to operate when it was best suited, then we could possibly have the best of both worlds. Such a method of automatic technique determination could potentially achieve results better than any single technique. With this as our goal, we set out to create something that could support switching between techniques on the fly.

It was decided that a framework must be designed that supports the simple inclusion of several techniques, and have them all play nicely with one another. The software would use software Interfaces, which allow the referencing of classes, without intimate knowledge of how they work, only that they use agreed-upon method names and general operation. With our selection techniques operating under these guidelines, we could easily drop them into the framework, achieving a type of "plug and play" experience that would make future expansion and re-use by others much simpler. Even if our framework was not used, the design strategies could be.

For the framework to operate, it requires that two or more selection techniques register with it, which is to say that they volunteer to participate in its operation. From there, the

framework has the responsibility of asking each technique how suitable they are at selecting, given the provided conditions. The technique that reports the best suitability will be chosen as the winning technique, and will be used if and when the user attempts to perform a selection.

For the selection techniques to make a decision as to their suitability, they must have information to work with. Since our previous study was based on object density and speed, we started there. We ended up using the number of objects within the cursor, the cursor velocity, and the velocity of the objects in the cursor. Other factors could have been considered, but at that point in our research, those three familiar attributes made the most sense to use.

Early in the design, a decision had to be made as to how to control the flow of information from the scenario to the selection techniques. One method is to gather the required information using the Analyzer (the core logic of the framework), and pass it into each technique. The second method is for each technique to measure the current conditions directly, which requires writing the technique in such a manner that it has access to the scenario in question. Since we had already chosen what attributes to read in, we chose the former technique, which also helped to isolate the techniques and streamline the flow of information.
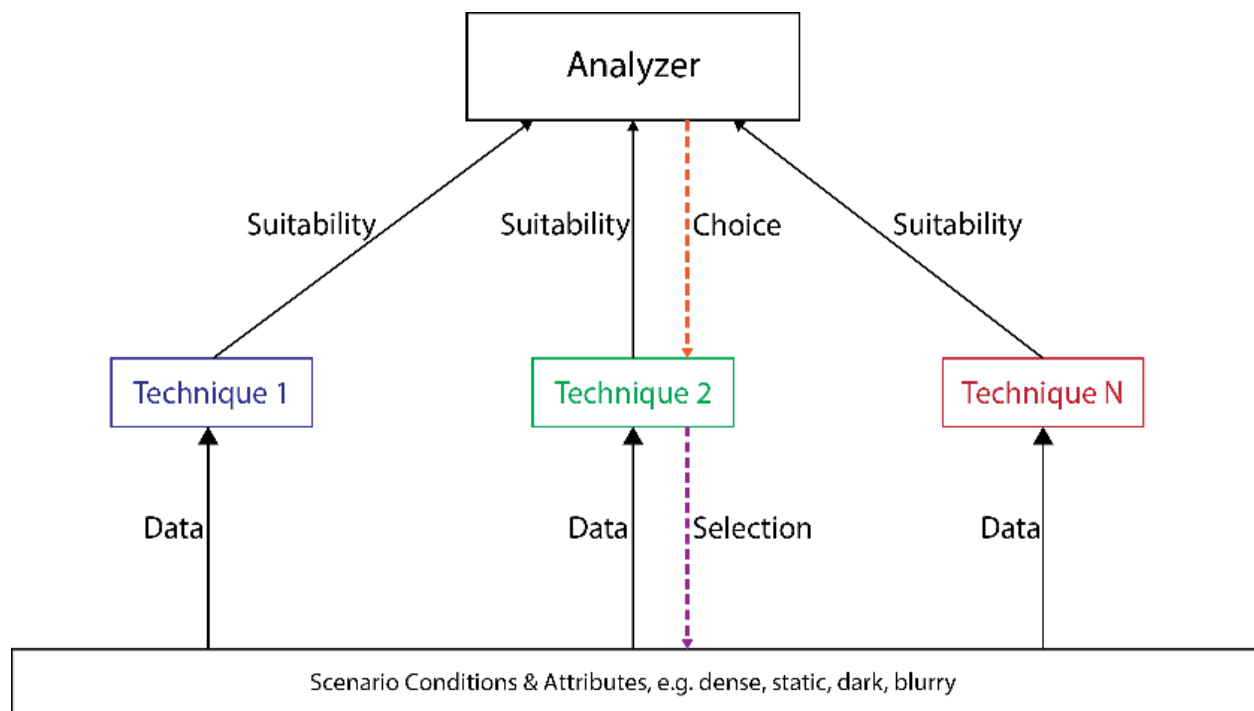
Figure 4.1: The Auto-Select Framework data flow

The utilization of the multi-selection-technique framework is called Auto-Select. This name reflects the fact that a collection of algorithms are at work, and the selection technique that is actually used when the time comes is automatically selected for the user without any explicit input.

We performed two evaluations of Auto-Select. We tested across three levels of object density and three different object velocities, which were designed to replicate the broad level of diversity commonly found in games and VEs. The object velocities ranged from nearly motionless to moving very rapidly across the screen. Each object travelled in a reasonably predicable manner. All of the objects were contained inside of an arena which was designed to give the user a good perception of the depth it contained. The first evaluation compared Raycast, Expand, and Auto-Select, where Auto-Select had

both to choose from. We had two variants of auto-select; the first one only considered cursor velocity and the quantity of objects within the cursor; the second variant additionally considered object velocities. Our second evaluation compared Bendcast (a variation of Spotlight), Expand, and a single version of Auto-Select that considered only object quantity and cursor velocity.

During the development and testing of our framework, we discovered several important factors that can significantly impact its effectiveness. The manner in which the framework is constructed plays a vital role in how the selection techniques can interact with the environment, and while we propose our framework, it is certainly reasonable that a developer may which to implement their own variation that gives them specific features they need that ours does not provide. From our data analysis, we also identified several characteristics of the auto-selection process that can introduce drawbacks which need to be addressed and minimized. In spite of the identified drawbacks, we believe that this method of optimal selection technique determination shows strong promise as an approach to improving 3D object selection.

*Section 4.1.1: Framework*

The core component of the framework is the Analyzer. Before a selection technique can be considered for use by the framework, it must first register with the Analyzer. Whenever desired, the software then asks the Analyzer for the optimal selection technique. This causes the Analyzer to poll all registered selection techniques, asking how suitable they are, given the provided conditions. Once all results have come back,

the optimal technique is then chosen and reported back to the software. Whenever a selection attempt is made, the software instructs the Analyzer to ask the currently optimal selection technique to perform a selection. While the technique is operating, the Analyzer is effectively suspended. When the technique is done, it reports back to the Analyzer, which in turn passes the results back to the software.

Each selection technique is required to implement the "ISelectionTechnique" interface. This allows the Analyzer to interact with it and perform specific tasks, such as determining the suitability index of the technique and commanding it to perform a selection. Determining the suitability index is performed by a method named, "getSuitabilityIndex". The parameters passed in to make this determination are described in further detail in section 4.2. It is in this method that the software developer would place their algorithm which takes in the conditions and establishes how suitable the particular technique would be at making a selection. To obtain quantitatively relative values, a developer should tune all related algorithms for all potential techniques using a standardized testing platform which best represents their game or simulation. In our design, the suitability values returned ranged between zero (not suitable) to one (very suitable). The software developer is free to normalize their values any way they like, such as using an estimated time required to select, or selection accuracy. As long as the values are all of the same type across all used selection techniques, there isn't a problem. The second method of large importance is the one that contains the actual selection algorithm(s). This is required by the interface so that the Analyzer can instruct the selection technique to perform the selection when desired. It also acts as a method

of returning the object selected back to the Analyzer and possibly other components of the software. We also included another important method which is responsible for informing the Analyzer what type of image the (if any) game engine should use to indicate that this selection technique is currently being chosen as optimal. This feedback is used to inform the user which technique they will be performing at any moment if they were to attempt to make a selection.

One important fact to note is that depending on which techniques are registered with the Analyzer, it is possible for one technique to completely supersede another. By this, we mean that it is entirely possible that technique M is always better than technique N, for any condition. In such a case, a user would rarely see technique N in use, and it would effectively not exist in the interface. This is not necessarily a drawback, but merely a consequence of using certain techniques. If this occurs, a developer can chose to simply leave out the weaker of the two techniques, or make modifications to either one so as to balance them better. A simple example of this would the Bendcast, which always selects an object more easily than Raycast, so it can be said that Bendcast completely supersedes Raycast.

*Section 4.1.2: Suitability Index Criterion*

When establishing an algorithm for determining the optimal selection technique, one has to extract information from the scene. Within the scene, there are many pieces of information that can be utilized. For our research, we focused on two: the number of objects inside of the cursor and the cursor velocity. These two were chosen due to their

primary importance when creating a taxonomy of selection techniques with respect to how they perform in dense and dynamic environments. For our first summative evaluation, we also included object velocities, and represented the inclusion of this third feature as a second auto-select technique. Other factors that were not chosen but could be incorporated include average distance to objects inside cursor, average size of objects inside cursor, level of occlusion, and more. It is entirely possible that a different set of conditions would yield better results, and it is up to the developer to determine which would be best, given the style of game or simulation they are developing.

We chose to control the data used to determine suitability by means of passing them in as parameters to each selection technique. Doing this enforces the requirement that the techniques use the same information when calculating their own suitability index. The drawback of this is that it requires the parameters to be decided ahead of time and built into the framework. An alternative method would be to simply allow the techniques to observe the world for themselves and select whatever data they like. This might give the developer more control, but reduces the flow of control that the framework has on selection. Ultimately, either way of handling criterion should be considered before implementation.

Deciding which information is most useful for determining the optimal technique can be difficult. Measuring the average distance to the objects in the cursor, combined with their size, would give a general idea as to how difficult each object is to select. This does not necessarily inform a user of how much of the object is inside of the cursor. If there were many average size objects within the cursor but far away, then the user

would experience nearly the same thing as if there were small objects close up to the user. In both cases, the objects appear small, and many fit within the cursor. To get a full understanding of the difficulty of selecting any of the objects, it would be best to consider the distance, size, and number of objects. Nearly the same level of understanding about selection difficulty can be obtained just by analyzing the number of objects in the cursor. For this reason, we chose to ignore object distance and size. In our software, we do pass in the list of objects currently inside of the cursor, so it would be possible for an algorithm to more thoroughly inspect the objects for such attributes, and possibly others not mentioned here.

Section 4.2: Baseline 3D Selection Techniques

To test our framework, we used three selection techniques: Raycast, Bendcast, and Expand. They were each tested individually and as a part of the framework. They were used in two different experiments, as described later. We chose these since they provided a wide spectrum of performance in different scenarios. We wanted to use only two techniques per study so that we could keep the complexity of each study to a manageable level. For the sake of simplicity, Raycast and Expand are not re-explained.

*Section 4.2.1: Bendcast*

The name "Bendcast" was derived from the fact that a ray used for raycasting is bent to hit a target, one which is closest to the center of the cursor. Only objects lying within the

cursor are eligible for selection. It may go by another name, such as Spotlight, but we designed it by building off of a simple Raycast technique and did not follow strict guidelines from other sources, so we gave it our own name to avoid confusion. This technique works best when only one object is located within the cursor, and becomes more difficult to use as the number of objects increases. From this technique, more complex techniques may grow. It featured a simple point and click experience that should be easily understandable to all users. Optimal Selection Technique Assignment

Our goal was to develop a method for determining the optimal selection technique across a broad range of scenarios. To do this, we created a flexible software framework that utilizes a primary Analyzer which interacts with one or more selection techniques to determine the optimal one, given any set of conditions. The accuracy of the Analyzer hinges on the accuracy of the independent algorithms within each technique. This framework was designed with the expressed intent of allowing 3rd party software developers to create their own selection techniques and plugging them into the Analyzer without any dependency issues. This gives the most amount of freedom and reduces the development time to just that which is required to perfect the selection technique. Our framework was developed by an iterative process where each method was carefully designed to give a high level of both functionality and flexibility.

*Section 4.2.2: Suitability Index Algorithms*

Each selection technique is responsible for having its own suitability index algorithm. This algorithm computes how suitable the particular selection technique is, given the

provided conditions. How the developer chooses to use this information should depend entirely on how that technique has been observed to perform in the expected situations. Key pieces of information should be utilized to make this decision, and also should be chosen based on observation and measurement. We developed two algorithms, one for each selection technique. They were developed iteratively and were a reflection of how we observed the techniques perform in our simulation.

For Bendcast, an increase in the number of objects and an increase in cursor velocity had a negative effect on its suitability. The suitability was inversely proportional to the number of objects, which was then negatively affected by a factor derived from the cursor velocity. A simplified version of the algorithm could be stated as 1 / (N × VC), where N is the number of objects in the cursor, and VC is the cursor velocity. In actuality, the suitability was very close for one or two objects, and shifted towards the stated simplification with three or more objects. The construction of this algorithm was based on one created for a basic Raycast technique. Raycast follows very closely to the simple formula, without any forgiveness for two or three objects. Knowing that Bendcast is more forgiving of the user, it was determined that slightly more than one object within the cursor had a lesser impact of selection difficulty.

For Expand, its suitability index actually increased as the number of objects and cursor velocity increased. There was an upper limit to the number of objects where Expand was no longer increasing in suitability. This is due to the fact that expand was only designed to fit 36 objects into its virtual grid at any one time (9 objects per row, 4 rows). After 36 objects, its suitability would start to decrease, since certain objects were forced

to be removed from the virtual grid. Should this happen, there is room for another technique to surpass it in suitability, perhaps a technique that is designed to work well for very large numbers of objects (greater than 36). A good candidate for this level of object density might be the SQUAD technique [35], which features an iterative reduction process and is suited to a large set of potential objects.

As a result of our inclusion of these selection techniques, Raycast and Bendcast were favored for automatic assignment when the cursor was relatively still and contained few objects. Once the user moved the cursor and / or several objects came inside of the cursor, Expand would be more likely chosen. This was the original intention, and was witnessed in actual use. While we desired to achieve an even split between intended use and automatic assignment, it was ultimately up to how the user performed in the simulation that dictated our real-word results.

*Section 4.2.3: User Feedback*

The algorithm that performs the auto-selection is very important, but another key component to the entire framework is how the user is informed that such a change of technique is taking place. If the user is not adequately informed of which selection technique will occur should they attempt to select, then they will likely not get the results that they expected. An unsatisfied user might just prefer to use a one size fits all technique and accept any shortcomings that may come with it. Thus, a great deal of emphasis should be put on how to inform the user that a switch of technique has occurred.

The method that we implemented was the design of a custom indicator icon, which was placed in the upper-right corner of the cursor (see Figure 4.2). Each selection technique had its own icon, and it gave a hint as to how the technique would function. This logical mapping was created with the intention of making it easier for the user to understand which technique would be used when they try to make a selection. For Raycast and Bendcast, our indicator was a hand with the index finger extended in a pointing pose. Additionally, a red laser was emitted from the tip of the finger, as if it was a laser pointer. For Expand, the icon featured a 3 × 3 grid of colored blocks, which represented the grid that the objects were placed in.
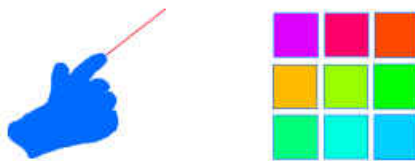


Figure 4.2: Feedback Indicators. Bendcast (Left) & Expand (Right)

To prevent the current technique from cycling rapidly, a delay was introduced into the Analyzer. We chose to use 500ms as the minimum time that any technique could be in use. If there wasn't such a requirement, then the user could experience rapid cycling, which would only serve to confuse and deter them. In our experience, a value as high as 1000ms would be suitable, and is ultimately up to the developer and their preference.

Section 4.3: Summative Evaluation One

To determine how well our Auto-Select algorithms (A and B) faired against the two static techniques (Raycast and Expand), we conducted a user study across three levels of

scene density and three levels of object velocity. We tested two versions of the Auto-Select technique. Both used cursor velocity and object quantity, but Auto-Select B also utilized the velocity of the objects within the cursor. Before performing the study, we established three hypotheses:

[H1]  Auto-Select will perform approximately as well as Expand.

[H2]  Auto-Select will perform better than the average of the two static techniques.

[H3]  Auto-Select B will perform better than Auto-Select A.

*Section 4.3.1: Subjects and Apparatus*

We ran 36 participants (29 male, 7 female), who's ages ranged from 18 to 29. These were all selected from the general student body of the University of Central Florida. The average participant claimed that they play games approximately once or twice a week. They also rank themselves as "average" with respect gaming skill. The entire experience for each participant took approximately 20 minutes, which included both a pre-questionnaire and post-questionnaire. Each participant was compensated $10 for their time.

Our system setup featured a 50" HDTV, An Intel Core-i7 Laptop with an Nvidia GeForce GTX 560M GPU, and a Sony PlayStation 3. The PlayStation 3 was utilized for its Move.Me SDK [52], which we used to capture the users input via the Move Controller, which is a 6-DOF tracking device featuring several buttons and is easy to use. 70% of participants reported that they had no previous experience with the Sony Move

Controller. The software used for development of our simulation was Unity 3.5, created by Unity Technologies [57].

*Section 4.3.2: Experimental Task*

Our participants were asked to perform selection tasks using the Sony Move controller. They were informed that they would be testing 36 difference scenarios that varied in number of objects and object velocity. For each scenario, a single purple object would be the desired target, and they were instructed to select it as quickly and efficiently as possible. The order of the scenarios was random and some may be difficult to perform. They were also told that they would be testing 3 different selection techniques. Each technique was described to them, and they were informed how to tell which one was being used for any single scenario. Before starting the trials, the participant was given sixty seconds to practice, which utilized the Auto-Select A technique as a way of giving them experience with all of the techniques. The underlying fact that there were actually two different auto-selection algorithms was withheld from them, since their differences are purely algorithmic in nature and appear nearly identical to the user. For each scenario, they were given 2 seconds to observe the scene and identify the target, as well as infer which selection technique would be utilized. Upon making a selection, they were played one of two sounds, one for correct selections and a different sound for incorrect selections. If a correct selection was made, the word, "Correct" was also displayed at the top of the screen. Upon completion of the 36th (final) scenario, they

were informed that the study was complete, at which point they proceeded to start the post-questionnaire.

The user was instructed to stand in the same approximate position as all other participants, which was roughly 6 feet from the display. They were told to hold the Move controller in any way that was most comfortable to them, and were given a demonstration of how it interacted with the screen. The trigger button on the back was their only means of selection, and all other buttons on the controller has no function, with the exception of the top buttons during practice, which would cause them to exit practice mode if they were comfortable with moving on. To prevent hand-eye mismatch [1], the controller projected a virtual pointer onto the screen which controlled the cursor.

*Section 4.3.3: Experimental Design and Procedure*

We used a 4 × 3 × 3 within-subjects factorial design where the independent variables were selection technique (including auto-selection algorithm variant) and scenario. The selection techniques included Raycast, Expand, Auto-Select A, and Auto-Select B. Scenarios included all nine variations of three different levels of object velocity and three different levels of object density, which is merely the number of objects in the scene. The dependent variables were selection completion time and number of incorrect attempts.

Each participant was read the overview of a standard consent form, and then asked to complete a pre-questionnaire (see APPENDIX C). Once that was complete, they were

brought over to the computer and given a detailed description of what they were to do in the simulation. Upon completion of the simulation, they were asked to fill out a post-questionnaire (see APPENDIX D). Once that was completed, that was the end of the study. All 36 participants yielded successful simulation trials, and as such all data was utilized for later analysis.

*Section 4.3.4: Scenarios Tested*

All of the scenarios tested were done within the same 3D arena. It was a room which was rectangular in nature and featured five walls. The users avatar stood at the end and looked into the area through were the 6th wall would have been. We tested three levels of object velocity and three quantities of objects in the scene, for a total of nine combinations. The quantity of objects was 100, 200, or 300. An example of the low density scenario can be seen in Figure 4.3, and high density can be seen in Figure 4.4. The average object velocity was 2, 4, or 6 meters per second. These values were derived by testing in pilot studies to obtain a reasonable level of diversity and give meaningful results. They can be labeled as slow, medium, and fast. The slow speed was nearly slow enough to be considered motionless. The type of motion performed by each object could be described as Brownian in nature, with an apparently random drift of direction that was somewhat unpredictable.

Figure 4.3: Low Density Scenario
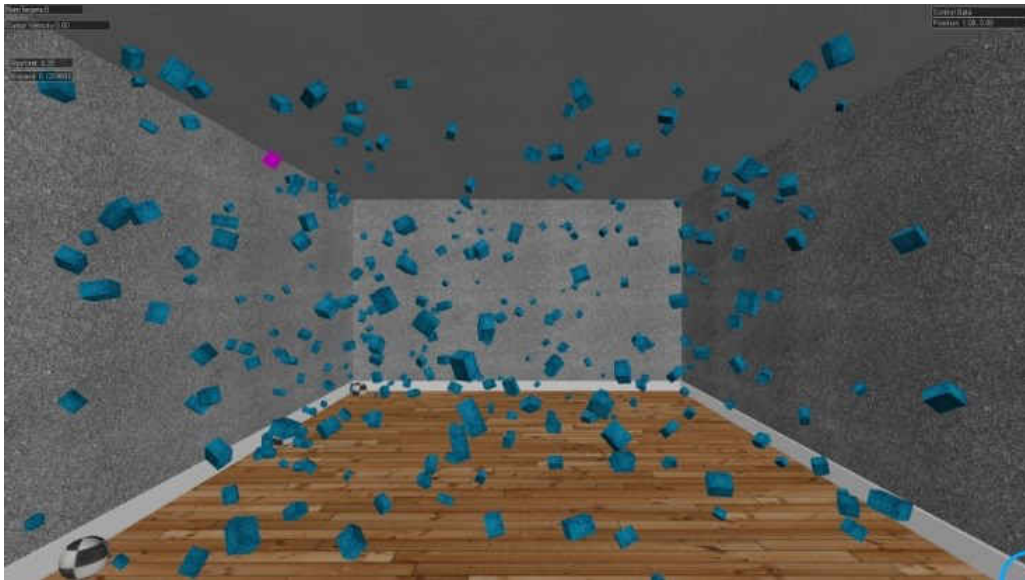


Figure 4.4: High Density Scenario

*Section 4.3.5: Experiment Results*

We analyzed our data by first creating linear regression models for the Raycast and Expand selection times as a function of object density and object speed. For Raycast and Expand, the models are:

$$R_t = (3.194 \; x \; [obj \; speed]) - (0.008 \; x \; [obj \; density]) - 3.759$$

$$E_t = (0.193 \; x \; [obj \; speed]) + (0.002 \; x \; [obj \; density]) + 1.729$$

Using ANOVA analysis, we found both models to be significant ($F_{2,316}$ = 49.4, p < 0.0001 for $R_t$, $R_2$ = 0.238) and ($F_{2,314}$ = 19.45, p < 0.0001 for $E_t$, $R_2$ = 0.11).

Using these models, we computed what the expected time would be for all 9 variations of speed and density for both Raycast and Expand. From these equations, we created a third Auto-Select technique, which we shall refer to as Auto-Select C, which simply chooses the technique which provides the minimum expected completion time. The result from Auto-Select C was then compared to the average for each scenario. Then the difference was computed, which represented how well our two auto-selection algorithms performed when compared to the data-driven algorithm. The average times for Raycast, Expand, Auto-Select A, Auto-Select B, and the computed time Auto-Select C are presented in Table 3. There were several data points that were outside of 4 standard deviations, so we removed them from our statistics.

Table 1: Average Completion time for (R)aycast, (E)xpand, (A)uto-Select (A), (B), and (C), in seconds.

| Density, Speed | R | E | AA | AB | AC |
|---|---|---|---|---|---|
| 100,2 | 1.91 | 2.35 | 2.02 | 2.03 | 1.83 |
| 100,4 | 4.12 | 2.49 | 2.57 | 2.26 | 2.70 |
| 100,6 | 19.64 | 3.28 | 4.49 | 3.38 | 3.09 |
| 200,2 | 2.66 | 2.41 | 2.58 | 2.60 | 1.03 |
| 200,4 | 4.36 | 2.82 | 3.43 | 3.38 | 2.90 |
| 200,6 | 11.95 | 3.16 | 4.54 | 4.91 | 3.29 |
| 300,2 | 3.00 | 2.75 | 3.26 | 3.15 | 0.23 |
| 300,4 | 3.10 | 2.82 | 3.29 | 3.19 | 3.10 |
| 300,6 | 15.21 | 3.4 | 7.27 | 4.77 | 3.49 |

In all but two cases, both of our Auto-Select algorithms do not perform as well as Auto-Select C (see Figure 4.5). This is expected, and can be explained by considering a few characteristics that aren't measured. First, the performance obtained by ASC is representative of an algorithm which would always choose the correct 3d selection technique, and represents what an algorithm could hope to achieve under ideal conditions. There would have to be absolutely no error, and the user would be fully educated on how to properly use the ASC technique. A primary factor that limits the accuracy and quality of any actual auto-selection algorithm is the performance of the user. No matter how good an algorithm is, the user may or may not be able to properly comprehend the idea that the 3D selection technique used at any moment could change. If they come to expect a certain technique when they press the button, then

they will behave as if it was always true. For example, if a user is expecting Raycast, they would probably take their time and try to get the desired object under the crosshairs. This remains true even if the technique going to be used is actually Expand. Some users are not aware of any feedback indicating that Expand is the current Auto-Select technique chosen, and thus will still act as if Raycast was going to be performed.

For the feedback mechanism we utilized, participants rated its usability a 4.7 on a 7-point Likert scale, with a standard deviation of 2. Based on this, there are obviously some improvements that can be made. One participant reported that they didn't even notice the feedback indicator, despite being instructed how it worked before the study. This was not a common response, but was note-worthy.

### Section 4.3.5.1: Overall

The average time taken per participant for all Raycast scenarios was significantly longer than the other three techniques, as illustrated in Figure 4.6. The standard deviation was also quite large, but this is not unexpected for Raycast. Some users preferred to trail moving objects for a while until they found a good opportunity to select, while others were very quick to react, thus achieving a shorter time, even if it meant making several selection errors. Expand, Auto-Select A, and Auto-Select B were all within statistical limits, and none were significantly faster than any other.

The average number of total errors per participant very closely resembles the average total time, as illustrated in Figure 4.7. Again, Raycast experienced significantly more

errors than the other three techniques. Expand, Auto-Select A, and Auto-Select B were all very close, but Expand did have fewer, which was statistically significant. There was no statistical significance between the errors of both of the Auto-Select techniques. We attribute this to the fact that their algorithms were similar in nature.
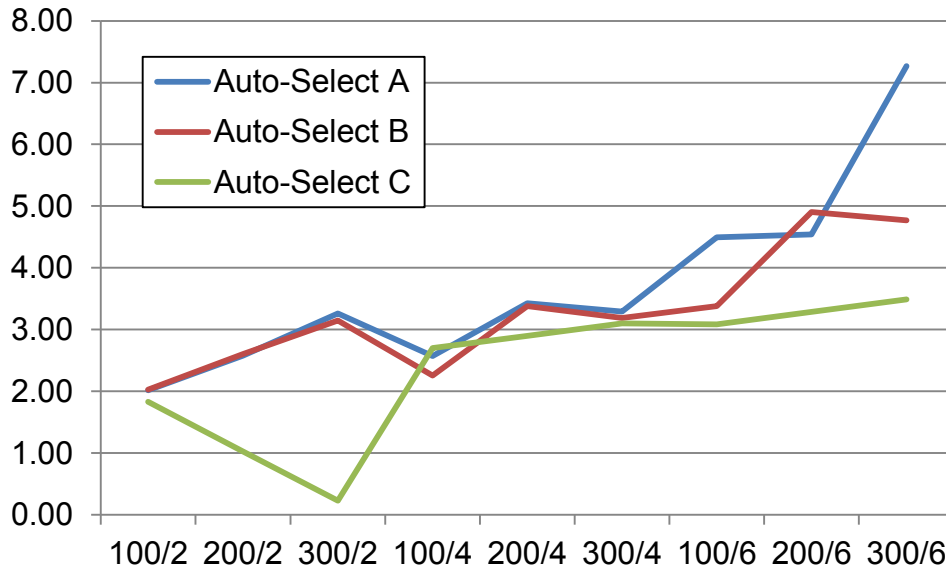


Figure 4.5: Performance Comparison between Auto-Select Algorithms
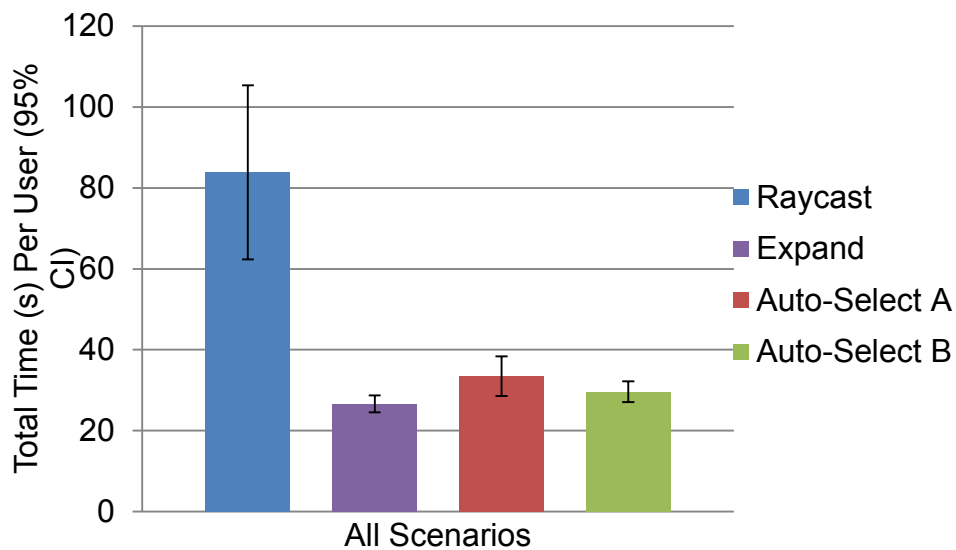


Figure 4.6: Mean Total Time, All Scenarios
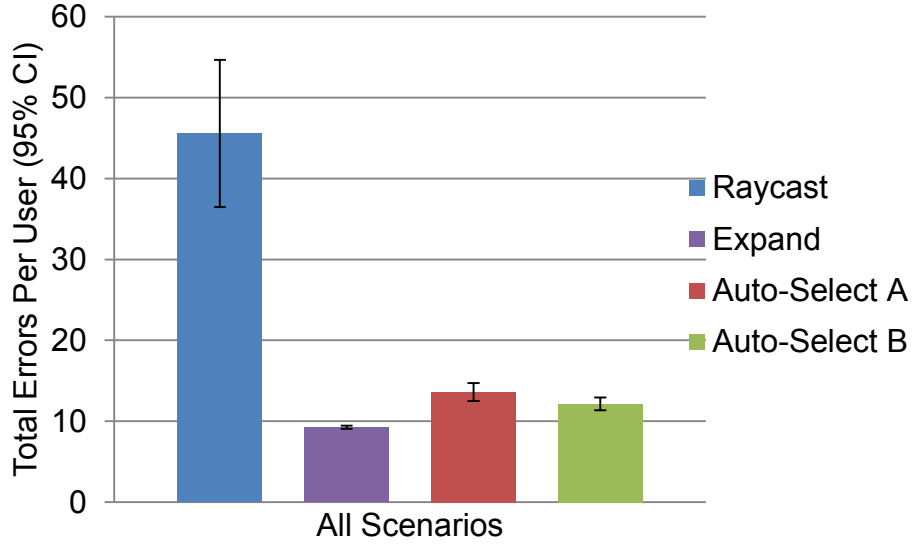
Figure 4.7: Mean Total Errors, All Scenarios

*Section 4.3.5.2: Post-Questionnaire*

Our Post-Questionnaire revealed that for the most part, users ranked Expand and Auto-Select with equal favorability (see Figure 4.8). Raycast was shown to be considered significantly less accurate and usable than both Expand and Auto-Select. This is not a surprise, since many participants verbally showed their dislike for it after finishing the trial portion. There was no statistically significant difference measured between Expand and Auto-Select in any sense. When ranking the selection techniques on overall favorability, Raycast was the least desired, while Expand was most desired. Auto-Select was a close second (see Figure 4.9).
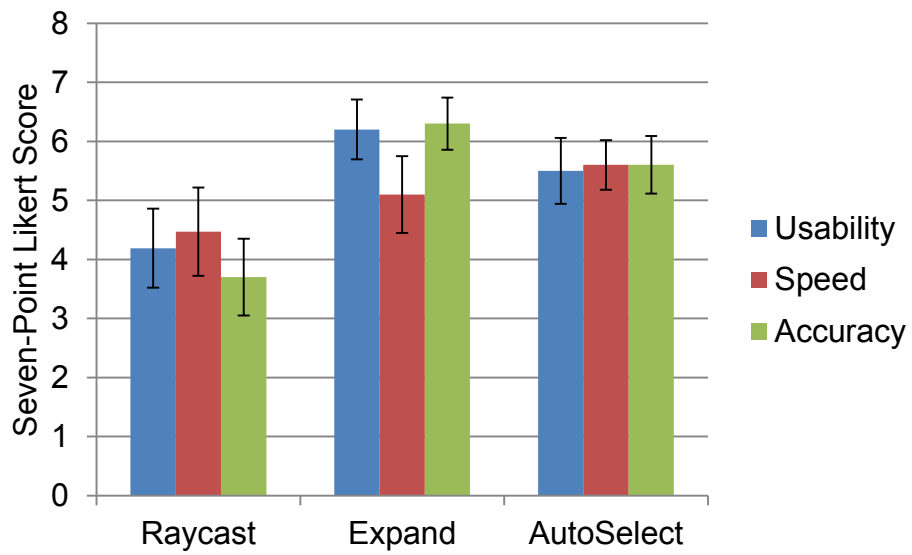
Figure 4.8: Post-Questionnaire, Technique Critique
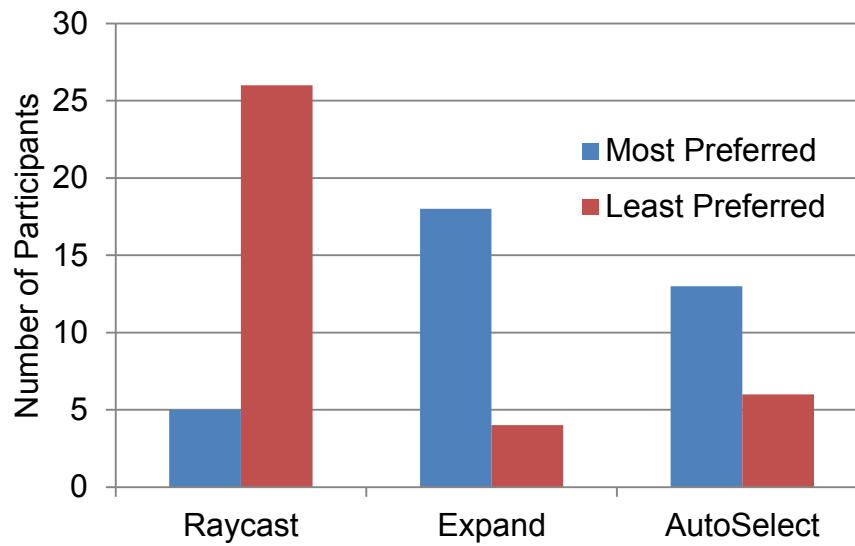


Figure 4.9: Overall Ranking of Selection Technique

*Section 4.3.6: Discussion*

As seen in Table 1, Expand took less time than Raycast in all 9 scenarios. Based on this observation, we can see that any Auto-Select technique would have been best served by picking Expand in all cases. However, this does not match up with the ASC algorithm though, which is based on statistical analysis of the performance of all participants.

When considering the performance of our two auto-selection algorithms, we must keep in mind that there will always be underlying factors that prevent the performance from always reaching the best of any possible technique. Due to these factors, we consider any auto-selection algorithm that is within 20% of the optimal time to be considered good. Based on this consideration, our auto-selection techniques could be considered, "good" in 9 of 18 scenarios. In one scenario, they both actually performed better than the data-driven ASC. Such an event would likely be eliminated if a higher number of samples were collected.

Being "in the zone" is a factor that we believe plays a role in performance, and will most likely impact the performance of an auto-selection technique in a negative way. When any 3D selection technique is currently in use, a user's mind acclimates to how they perceive it to act and respond to their input, and will come to expect certain behaviors from it. When the technique changes, there is the possibility that the users concentration will be reduced, thus causing a sense of confusion [63].

Another factor that negatively affects performance is the act of switching techniques after the user has already started the mental process of performing a selection. To counteract this, the Analyzer could be designed to revert back to the previous selection technique if a selection attempt is made within N milliseconds of switching techniques. At some value of N, it can be assumed that the user did not intend on using the "new" optimal selection technique, so it should not be called upon. Such a modification to the Analyzer would be minor and easy to implement.

The quality of the feedback mechanism also plays a large role in the user's ability to understand which technique is currently active. This will have a negative effect on the required time in two ways. The first way is due to any vagueness or insecurity about what will happen when they try to select. The user might be very cautious and perform as if the least forgiving technique was active, in an attempt to minimize errors. The second way is just strictly based off the fact that the user may incorrectly identify which technique is active. This type of error can be attributed to a feedback technique that is difficult to see or hear, not distinct between different techniques, or just plain unintuitive. If any such problems are present, then the user stands little chance of being effective with any auto-selection technique, no matter how well designed it is.

An interesting observation that we made was how little of an impact the scene density had on total selection time for both Raycast and Expand. For Raycast, the selection times generally went down as density increased. This seems counter-intuitive, but is backed up by the data. This is most noticeable when the velocity was high. Due to this, the model which describes the predicted selection time for Raycast actually assigns a

negative correlation between the number of objects in the scene and selection time. This inadvertently causes the ASC algorithm to predict a relatively low selection time for high density, low velocity.

For hypothesis 1, we predicted that Auto-Select would perform about as well as Expand. Based on Figure 4.6, there was no statistically significant difference between Expand and both Auto-Select techniques. Therefore, we state that hypothesis 1 is true. To further strengthen this claim, the qualitative data from the post-questionnaire (see Figure 4.8) shows that the participants also rated Expand and Auto-Select [as a whole] the same, with no statistically significant difference. Based on this, we know that there is no major deterrent to further use of an auto-selection technique, as both the quantitative and qualitative data support its place in the field of selection.

Hypothesis 2 is similar to [H1], but includes reference to the Raycast technique. In our studies, Raycast perform relatively poorly, and since [H1] is true, then [H2] will be true. Had Raycast performed better than Expand then this would not have been an automatic truth as is the case.

Our third hypothesis is very close to being true, but there is no statistical significance to back it up. Auto-Select B does perform slightly better than Auto-Select A with regard to both total time and total errors, but not significantly so. It is because of this that we must state that hypothesis 3 is incorrect, but close to valid. We believe that with a stronger difference between the two algorithms, we would have seen an even larger improvement in the performance. The small gain we did see lends itself to the idea that this still more performance left on the table for future developers to achieve.

## Section 4.4: Summative Evaluation Two

After seeing the results from our first summative evaluation, we decided to make some adjustments. To combat the domination that Expand had over Raycast, we built the Bendcast technique, which, as we described earlier, works much better than Raycast in virtually all cases. Also, we dropped Auto-Select B, and just stuck with the basic Auto-Select A, since their performance was comparable. The scenario configurations remained the same, and the general guidelines were also similar. For clarity, we will re-explain the specifics, as they are slightly different in some cases.

### Section 4.4.1: Subjects and Apparatus

We ran 27 participants (19 male, 8 female), who's ages ranged from 18 to 27. These were all selected from the general student body of the University of Central Florida. The average participant claimed that they play games approximately once or twice a week. They also rank themselves as "average" with respect gaming skill. The entire experience for each participant took approximately 30 minutes, which included both a pre-questionnaire and post-questionnaire. Each participant was compensated $10 for their time.

Our system setup featured the same display, computer, Sony PS3, and Sony Move SDK as the first evaluation. The software was the same, with the exception of the changes required to make the modifications described at the beginning of the section.

## Section 4.4.2: Experimental Task

Our participants were asked to perform selection tasks using the Sony Move controller. They were informed that they would be testing 27 different scenarios that varied in number of objects, object velocity, and selection technique. For each scenario, a single purple object would be the desired target, and they were instructed to select it as quickly and efficiently as possible. Each scenario was performed five times, for a total of 135 selection scenarios. The order of the scenarios was randomized to prevent any bias due to gained experience during the trial. Each technique was described to them, and they were informed how to tell which one was being used for any single scenario. Before starting the trials, the participant was given sixty seconds to practice, which utilized the Auto-Select method as a way of giving them experience with all of the techniques. For each scenario, they were given 2 seconds to observe the scene and identify the target, as well as infer which selection technique would be utilized. Upon making a selection, they were played one of two sounds, one for correct selections and a different sound for incorrect selections. If a correct selection was made, the word, "Correct" was also displayed at the top of the screen. Upon completion of the final scenario, they were informed that the study was complete, at which point they proceeded to start the post-questionnaire.

The user was instructed to stand approximately six feet from the display. They were told to hold the Move controller in any way that was most comfortable to them, and were given a demonstration of how it worked. The trigger button on the back and the move button on the front were their only means of selection. The four symbol buttons on the

79

top were programmed to terminate the practice session if the participant desired to do so. To prevent hand-eye mismatch [1], the controller projected a virtual pointer onto the screen which controlled the cursor.

### Section 4.4.3: Experimental Design and Procedure

We used a 3 × 3 × 3 within-subjects factorial design where the independent variables were selection technique (Auto-Select, Bendcast, Expand), scene density, and object velocity. Three levels of object velocity and three levels of object density were tested, for a total of nine possible scene configurations. The dependent variables were selection completion time and number of attempts.

Each participant was read the overview of a standard consent form, and then asked to complete a pre-questionnaire (see APPENDIX E). Once that was complete, they were directed to the computer and given a detailed description of what they were to do in the simulation, as well as how to use the input device. Upon completion of the experiment, they were asked to fill out a post-questionnaire (see APPENDIX F). Participants were reminded which technique was which, and given the opportunity to inquire about anything on the post-questionnaire. Once that was completed, they were formally done with the study. All 27 participants yielded successful simulation trials, and as such all data was utilized for later analysis.

*Section 4.4.4: Scenarios Tested*

All of the scenarios tested were identical to the previous evaluation. We tested with 100 (see Figure 4.3), 200, and 300 (see Figure 4.4) objects per scene. The average object velocity was 2, 4, or 6 meters per second.

*Section 4.4.5: Experiment Results*

Each of the 27 scenarios was completed 5 times by each participant, and then the average of the 5 runs was used for all further analysis. An in-depth study of the results yielded some interesting results, both quantitatively and qualitatively.

*Section 4.4.5.1: Overall*

The average time per user per scenario for each technique is shown in Table 2. Based strictly on these times, Bendcast comes away as the fastest. For Bendcast, it was always the case that an increase in density or speed caused an increase in required time. This is also true of Expand, with the exception of medium density and speed, which is likely an anomaly. Like Bendcast, Auto-Select also experienced an increase in time as either density or speed increased. For total time spent using each technique, Auto-Select was significantly faster than Expand ($t_{26}$ = 5.52, $p < 0.01$). Likewise, Bendcast was significantly faster than Expand ($t_{26}$ = 29.07, $p < 0.01$) and Auto-Select ($t_{26}$ = 14.5, $p < 0.01$). Figure 4.10 reveals the average total time required for a participant using each technique, e.g. each participant took an average of 13 seconds to

make their selections using Bendcast when all 9 speed/density scenarios are combined. These values represent multiplying the average values from Table 2 by nine [scenarios]. It is worthy to note that there was no penalty incurred for making an incorrect selection, so these times do not reflect any inherent penalties that might be experienced in an actual game or simulation.

Error data is presented in Table 3. Each value represents the average number of errors made by a single participant in a single scenario using the indicated selection technique. Contrary to the positive results for Bendcast, it experienced the highest error rate of all, regardless of speed or density. Bendcast experienced significantly more errors than Expand ($t_{26}$ = 13.4, $p < 0.01$) and Auto-Select ($t_{26}$ = 7.94, $p < 0.01$). Expand did manage to experience significantly fewer errors than Auto-Select ($t_{26}$ = 7.28, $p < 0.01$). With one exception, Expand was always the most accurate. Auto-Select had a relatively low average error rate, just double that of Expand and about two fifths that of Bendcast. From Figure 4.11, we can see that Bendcast did experience significantly more errors than either of the other two techniques. In our testing, this didn't impact the user in a negative manner.

When the time and error results are combined, some observations can be made. Since Expand is a two-step process, it makes sense that it took more time to select than a simple single-step technique such as Bendcast. By extension, Auto-Select exhibited a similar trait, since it included Bendcast in its portfolio of available techniques to choose from. The fact that Auto-Select chooses between the two technique, it only makes sense that its results were somewhere in between for both time and errors. With perfect

algorithms for determining when to use either technique, it could be expected that Auto-Select would be both faster and more accurate than any single technique in its arsenal. In actuality, the algorithms for determining suitability are not limited to only optimizing just speed or accuracy, but must take a holistic approach and find the most balanced means of optimizing both without making any significant sacrifices.

An important fact to keep in mind is that there was no penalty for an incorrect selection. As a result, there was no natural tendency to play it safe. In a real simulation or game, it is not uncommon to see the user require several seconds to undo an invalid selection. Not only would this have an inherent negative effect on selection time, but would also have an impact on how careful the user is when making their selection. The result of this would cause our two chosen techniques to have more similar selection times, and thus increase the impact that the Auto-Select algorithm could have on overall selection quality.
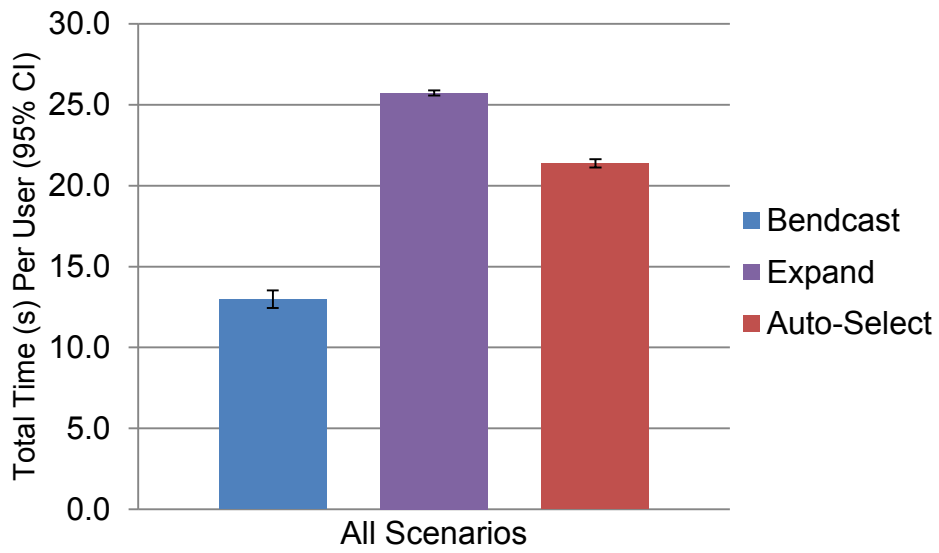
Figure 4.10: Mean Total Time, All Scenarios
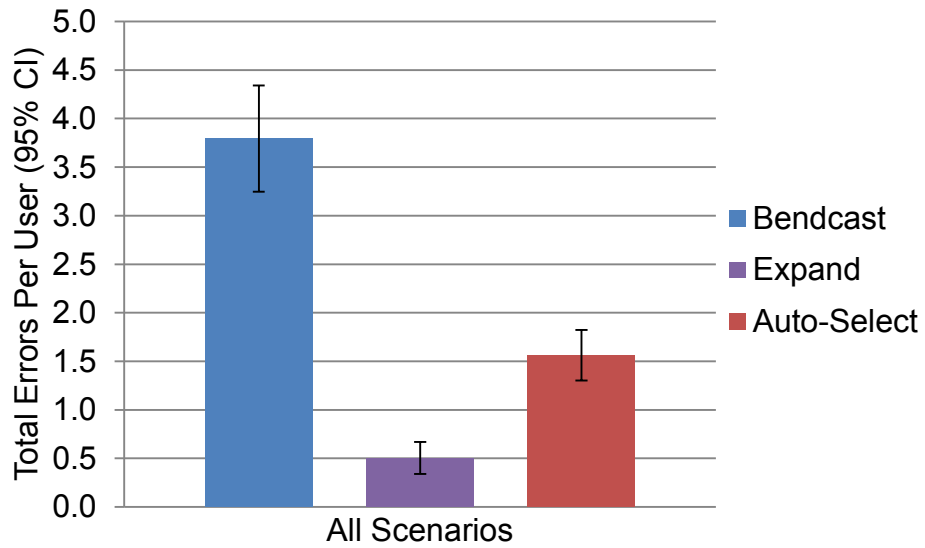


Figure 4.11: Mean Total Errors, All Scenarios

Table 2: Average Completion time for (B)endcast, (E)xpand, and (A)uto-Select

| Density, Speed | B | E | A |
|---|---|---|---|
| 100,2 | 0.87 | 2.27 | 1.65 |
| 100,4 | 1.06 | 2.48 | 1.87 |
| 100,6 | 1.48 | 3.14 | 2.34 |
| 200,2 | 1.05 | 2.64 | 1.88 |
| 200,4 | 1.40 | 2.46 | 2.44 |
| 200,6 | 1.70 | 3.27 | 2.82 |
| 300,2 | 1.21 | 2.69 | 2.50 |
| 300,4 | 1.78 | 3.23 | 2.57 |
| 300,6 | 2.45 | 3.55 | 3.32 |
| *Average* | *1.44* | *2.86* | *2.38* |

Table 3: Average errors for (B)endcast, (E)xpand, and (A)uto-Select

| Density, Speed | B | E | A |
|---|---|---|---|
| 100,2 | 0.08 | 0.21 | 0.07 |
| 100,4 | 0.17 | 0.03 | 0.10 |
| 100,6 | 0.29 | 0.07 | 0.13 |
| 200,2 | 0.21 | 0.05 | 0.09 |
| 200,4 | 0.47 | 0.00 | 0.23 |
| 200,6 | 0.53 | 0.07 | 0.32 |
| 300,2 | 0.36 | 0.02 | 0.07 |
| 300,4 | 0.77 | 0.08 | 0.21 |
| 300,6 | 0.91 | 0.16 | 0.34 |
| *Average* | *0.42* | *0.08* | *0.17* |

*Section 4.4.5.2: Post-Questionnaire*

A study of our Post-Questionnaire showed that participants ranked Bendcast and Auto-Select very close in favorability (see Figure 4.12). Participants rated Expand and Auto-Select more accurate than Bendcast. This qualitative result closely resembles the actual performance as shown in Table 3. Expand and Auto-Select were also rated as more usable than Bendcast, slightly. Surprisingly, participants did not seem to think that there was any significant difference in the speed between the three techniques.

For the feedback mechanism we utilized, participants rated its usability a 5.0 on a 7-point Likert scale, with a standard deviation of 1.8. Based on this, there is some room for improvement. One participant reported that they didn't even notice the feedback indicator, despite being instructed how it worked before the study. This was not a common response, but was note-worthy. Overall, it was a worthy feature that did prove beneficial to most.

The importance of ease-of-use and speed were both rated a 6 out of 7 on a Likert scale, with a standard deviation of 1.1 and 1.2, respectively (see Figure 4.13). Participants were also satisfied with the amount of practice time given, with a Likert score of 6.8 and standard deviation of 0.58, the highest of any response on the questionnaire.
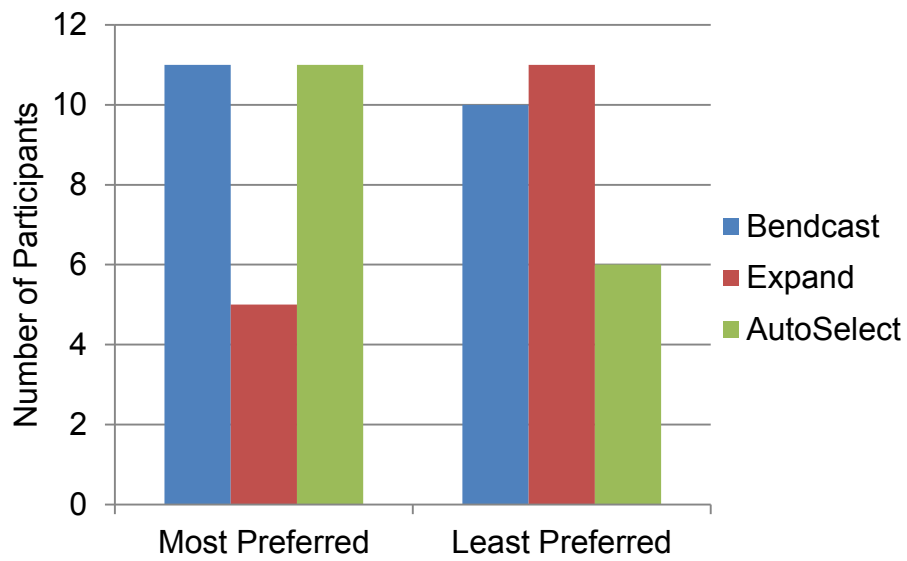
Figure 4.12: Overall Ranking of Selection Technique
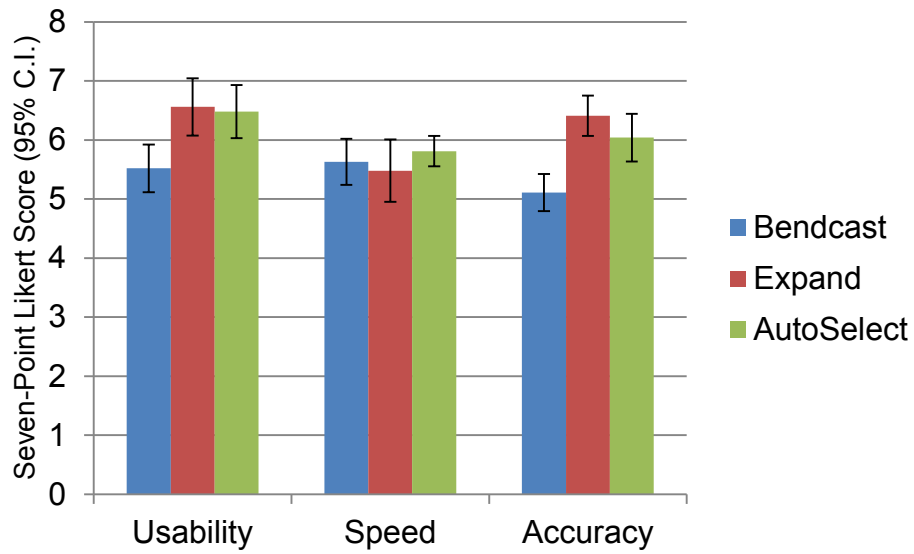


Figure 4.13: Post-Questionnaire, Technique Critique

*Section 4.4.6: Discussion*

Contrary to the first evaluation, Expand was always the slowest technique (see Table 2). While slower, it always experienced fewer errors that Bendcast (see Table 3). Both of these results must be considered in a balanced manner to determine how to appropriately tune the Auto-Select algorithms. Based on our results, it is obvious that once again, we inadvertently chose a combination of techniques and scenarios where one technique is always the best. Since Expand is geared more towards denser environments, we needed to be testing with more objects on the screen to shift the balance of power more towards Expand.

Another way to look at the results is to consider the lack of penalties for incorrect selection, besides time already consumed. If we imagined a system that displays a secondary menu once an object is selected, then an incorrect selection would require the user to indicate that they need to "go back", and try their selection again. This inadvertent time penalty would cause techniques that are fast but error-prone to suffer much more than those that are perhaps slower, but much more accurate.

Our desired result would have been to see the Auto-Select algorithm perform better than either technique did independently. This was obviously not the case, but several critical reasons were discovered that lend an explanation as to why. It is only after 14 hours of user trials did we discover the full extent of how our chosen techniques would perform together. Addressing the issues found in our observations as well as a more thorough testing cycle could lead to more fruitful results in future studies.

Another key research contribution of this work was the design of the dynamic environment. Within it, we were able to dynamically configure the quantity and velocity, and visual specifics of the objects at any time. This gave us the flexibility to create a testing framework that could utilize any level of object density and level of activity (dynamic component) so that we could test anything we needed to. In our case, we did a 3x3 test design, but could have easily done more or less with no issue at all. In addition to that, we could have also dynamically adjusted properties of objects in mid-scenario and tested how participants handled that as an entirely different research objective. This fine grain control ultimately gives us the power to accurately classify techniques within the scope of the configurable conditions, without having to make do without any specific data point. Our second research question is satisfied by our environment, and is improved further in our work in future chapters as it gets refined.

With the data obtained from our first two research contributions, we have established an initial understanding of technique classification based on our desired attributes, as well as an initial attempt at programming this knowledge into an algorithmic framework that determines an optimal technique under given conditions. Although our results with the framework were not amazing, they were still a step in the right direction with regard to matching techniques to their environment via software, which answers our third research question. We later expand on this idea in our remaining chapters as we look more into dynamic techniques that respond to varying conditions on the fly.

## Section 4.5: Summary

Our proposed framework has been shown to be capable of handling the task of considering several selection techniques without regard to how they are written, as long as they implement the correct software interface. This flexibility can and will allow software developers to easily reference their own selection techniques and take advantage of our framework to enhance their own simulations and applications. Our initial exploration into this area of research revealed weaknesses in our initial understanding and opens the possibility of further improvements in several areas, such as algorithms and framework design. Our chosen criterion showed to be on the right step towards optimally making this determination, and will give future developers a head-start in developing their own suitability index algorithms.

In an ideal situation, an Auto-Select framework could decide the best technique to use all the time, but we believe that our data shows that this is likely a very hard goal to achieve. Like many other aspects of simulations and games, the fine tuning required to improve accuracy in such algorithms would rely on a thorough amount of testing and change. There are so many factors that influence selection difficulty, and they can vary wildly with the interface. It is even the case that these factors change depending on what part of the simulation or game the user is in. From here, we look to identify selection techniques that can be tied together to work best as a pair, and focus less on being able to take any two and mash them together. This bonding will establish a single technique that acts as a hybrid technique, with performance benefits tightly interwoven, contrary to the Auto-Select framework which treats the techniques separately.

# CHAPTER 5: DYNAMIC ADAPTATION OF 3D SELECTION TECHNIQUES FOR SUITABILITY ACROSS DIVERSE SCENARIOS

## Section 5.1: Introduction

In our previous chapter, we studied the effectiveness of the Auto-Select framework, and made some realizations about how it could have worked better. The transition between techniques is going to be critical in preventing the user from becoming confused about what is happening. For it to really work, each technique must seamlessly transition into the next, without even being noticed. We had attempted to present the user with a form of visual feedback, an icon, to represent which technique was active at the time. Through our user study, we learned that users often did not even notice what the icon was. Because of these challenges, we explored another way to possibly achieve the same goal of improved performance while still joining multiple techniques.

Instead of starting with two techniques and joining them together via switching, we attacked the problem from the other direction. We started with a single technique, and integrated the functionality of another technique into it. At its core, there was only ever one technique, thus no switching required. The logic that was once a part of the Auto-Select Analyzer was now embedded into the single technique, giving it tighter control over how to operate in any given set of conditions. What was previously a discrete transition was now an internal switch of operation that could be performed gradually and with more control.

To establish this new dynamic technique, we explored the existing field of techniques and identified a few that had features that we believed could be modified and combined to create a new technique, one with the flexibility of working well across a broad range of scenarios. Our new technique is called Scope, and it was inspired by DynaSpot [14], Hook [45], and IntenSelect [20]. From DynaSpot, the idea of dynamically adjusting the size of the activation area was adopted. Unlike DynaSpot, we also adjust the size of the visible cursor and designed it to minimize occlusion. Many other design differences were implemented and are described in Section 5.2:. From IntenSelect and Hook, we utilized the distance-based scoring concept for approximating which object is likely desired by the user. However our specific method of performing this differs.

In order to test Scope, we performed a user study using a test bed that featured a sealed room with floating objects resembling water molecules. Nine possible scenarios were tested, including three levels of object density (number of objects in the scenario) and object velocity. We compared Scope to a traditional Raycast, as well as Bendcast [12] and Hook [45]. The dependent variables were completion time and number of errors. The results showed that Scope performed on par with Bendcast and Scope, yet was chosen as the most desirable technique by a significant margin.

## Section 5.2: Scope: Our New Selection Technique

Scope was created by implementing our own variation of several existing methods and combining them into a selection technique that is capable of operating effectively in

different scenarios. Below we describe our design goals and the distinguishing characteristics that set Scope apart.

*Section 5.2.1: Scope: Design Goals*

We designed Scope with the goal of having it work well across a broad range of conditions. To achieve this, we utilized ideas from different selection techniques and tried to make them all work well together. We were already familiar with fundamental techniques like Raycast and Spotlight, as well as more modern ones such as DynaSpot [5], Starfish [18], and Hook [14]. What we needed to do was identify key features and strong points that could be incorporated into a single technique. Ultimately, we chose to implement the speed-dependent behavior from DynaSpot for its ability to vary between a point-cursor and an area-cursor. We also chose to implement the distance-based scoring algorithm similar to Hook, since it provided a proven method of identifying targets that are most likely desired by the user. With these features, we then branched out and made some modifications and introduced some new ideas to improve the potential of Scope.

*Section 5.2.2: Speed-Dependent Behavior*

There are conditions where Raycast operates better than Spotlight, and visa-versa. Additionally, targeting an object is generally more difficult when the object is moving [3]. To overcome this problem, DynaSpot was created with the ability to dynamically adjust

the activation area (spot) of the cursor. With the cursor motionless, the spot would reduce to a single point, thus causing the technique to operate like Raycast. As the cursor begins to move, there is an initial ramp up period where the spot grows, not exceeding a certain maximum size. If the cursor slows to a stop, the spot shrinks after some time. This speed-dependent behavior was shown to give DynaSpot an 18% performance advantage over Raycast. Because of that, we chose to implement our own modified version. This speed-dependent zooming behavior is also seen in a 2D context [32].

We took this idea of speed-dependent behavior and implemented a version that had a few key differences. DynaSpot permitted their spot size to decrease to just 1, thus becoming Raycast. In our own testing, we observed that maintaining an area of more substantial size was beneficial. Our function for computing the size of the cursor is

$$\textsc{SpotSize} = \textit{Clamp}(\textsc{CursorVelocity} \times \alpha, \textsc{S}_{\textsc{MIN}}, \textsc{S}_{\textsc{MAX}})$$

where $\alpha$ adjusts the sensitivity to the user input, and SMIN and SMAX are the minimum and maximum sizes that the spot can be, respectively. The size of the cursor when smallest was 25% the size of when it was largest. We performed manual testing to determine suitable minimum and maximum sizes.

To control our spot growth functions, we leveraged existing input filtering which is used to smooth the user input. To compute the cursor velocity at any moment, we sampled the cursor position and stored it for the past fifty frames. From this, the distance and time was measured between the points to evaluate what the average velocity was over

94

that range. The weight placed on each sample was equal, which caused the computed velocity to lag somewhat behind its true instantaneous speed. This benefited us by providing a natural ramp-up delay in the size of our spot, as well as a ramp-down that was equally favorable. We were able to achieve similar sizing behavior of DynaSpot by implementing our algorithm in this manner.

The obvious visual distinction between Scope and DynaSpot is that Scope does also adjust the visual size of the cursor, not just the activation area. With DynaSpot, the activation area changes in size, but the cursor does not. This was done to prevent an oversized cursor from obscuring relevant screen information. This can be a potential problem if there is no other way in which the user is informed of the enlarged activation area. To avoid the issue of cursor-object occlusion, we utilized a circular cursor that is mainly comprised of a thin outer ring with and a small cross-hair center (see Figure 2). This gives the user the impression that anything inside of the cursor is fair game. The outer border itself is also translucent, thus allowing even small objects the chance to be seen. Additionally, objects behind the border are not eligible for selection, and thus the small amount of occlusion is not likely to pose a serious threat to the user's ability to make their desired selection.

*Section 5.2.3: Nearest-Object Determination*

Like a typical Spotlight technique, DynaSpot resorts to selecting the object closest to the center. A different approach is taken by techniques such as IntenSelect and Hook, which use scoring algorithms to determine which object has, on average, been closest

to the center of the cursor over time. With Hook, all objects are measured, but only a limited number of objects can have their score adjusted. A somewhat complicated set of rules then determine how the scores of the objects move up or down. Our approach is more direct. At all times, Scope is aware of which targets are located inside of the cursor. From this set, the distance is measured and stored for reference. For each frame, the previous 0.5 seconds of distance data is used for each object, and from that, the average distance is computed. The object with the lowest average distance is declared the winner, and gets highlighted in such a manner that indicates to the user that it would be chosen if a selection attempt was made.

*Section 5.2.4: Summarization*

In summary, Scope differentiates itself from existing techniques in the following ways:

•       Inclusion of dynamic cursor resizing that matches dynamic activation area resizing 1:1

•       Minimum cursor/activation size remains circular, versus single point as in DynaSpot

•       Distance-sampling of near-by objects only occurs within the bounds of the cursor, not entire scene

Section 5.3: Summative Evaluation

We conducted a user study to evaluate the performance of our selection technique. We compared the performance of Scope against Raycast, Bendcast, and Hook across three levels of scene density and three levels of object velocity. Raycast was chosen as a

baseline technique. Bendcast and Hook were chosen for their similarity in features to Scope.

*Section 5.3.1: Subjects and Apparatus*

We ran 27 participants, 22 male and 5 female, who were between 18 and 29 years of age. Participants were solicited from The University of Central Florida. Approximately 37% of users reported previous experience with the Sony Move Controller, and the average respondent reported that they played 3D games once or twice a week. Each participant took approximately 30 minutes to complete the study, and was compensated $10.00 for their time. Before the experiment, they were asked to complete a questionnaire to gauge their general experience with gaming. Afterwards, they were presented a post-questionnaire to gauge their opinion on the tasks they were asked to complete.

Our test configuration included an Intel Core-i7 laptop with 16GB of RAM, a GeForce GTX 560M 3GB GPU, and a 55" FHD LCD display. The input device was a Sony Move controller, powered by the Sony Move.Me SDK on the Playstation 3 [16]. We used Unity 4.2 to power the simulation [17].

*Section 5.3.2: Experimental Design and Procedure*

We utilized a 4 x 3 x 3 within-subjects factorial design, with the selection technique (4 total), scene density (low, medium, high), and object velocity (slow, medium, fast) as

independent variables. The four selection techniques tested were Raycast, Bendcast, Scope, and Hook. The dependent variables were completion time and total number of attempts. For each scenario, the user was asked to select the indicated object. The total time measured for each scenario included all attempts, even incorrect ones. The timer would start when the user was permitted to begin the selection task, and would end upon selection of the correct object. The timer did not stop in-between attempts, and no explicit time penalty was awarded.

Each participant was presented with a consent form, and given the opportunity to read it and ask questions. Once consent was given, the participant was asked to complete a pre-questionnaire, which featured 7 questions, and took approximately one minute to complete. Once completed, the participant was brought to the testing area, and given a recited set of instructions for how to use the input device and what tasks they would be asked to perform. The user would then be permitted to start the experiment. Upon completion of the experiment, the participant was asked to complete a post-questionnaire which asked for their opinions on how they perceived the quality of the experiment, as well as how they rated various attributes of the selection techniques tested.

*Section 5.3.3: Experimental Task*

Each participant was informed that they would be performing a series of selection tasks. Their primary goal in each task was to select a specific object, which was indicated by the fact that it was the only green object in the scene. Each combination of selection

technique, scene density, and object velocity was tested by each user, a total of five times. In total, each participant performed 180 selection tasks. The order of the tasks was randomized for each participant, so as to minimize any skew from learning while performing.

The participant was given and shown how to use the Sony Move Controller. They were allowed to use which ever hand they desired and could hold it any way they wanted. All participants were instructed to stand in a particular spot, so as to fix the distance to the screen. Each participant was given an explanation of how the software would work, and what to expect. They were instructed to select the target object as accurately as possible.

Before a participant began the measured portion of the experiment, they were given sixty seconds to practice the selection techniques first, without any consequences. Users could terminate the practice session at any time via the press of a button. Each scenario began with the user able to see the scene and could begin selection at the same time. The object which would be selected if an attempt was made was highlighted to inform the participant of this fact. After each selection attempt, the user was given an audible feedback, a pleasant sound for a correct selection and an unpleasant sound for an incorrect selection. If a selection attempt was correct, the task was completed, and a message was displayed indicating that their selection was correct. This message was displayed on the screen for two seconds. If an incorrect selection was made, they simply continued with their attempts to select the correct object. After each correct selection and notification, the user was then given the next task. This proceeded until all

36 selection tasks had been performed. At that point, the software would be restarted and the process was repeated, until it had been performed a total of five times. After the fifth iteration, the entire experiment was complete. The participant was then asked to complete a post-questionnaire which contained questions designed to get their opinion of the techniques and the experiment as a whole.

*Section 5.3.4: Experiment Results*

Each of the 27 scenarios was completed 5 times by each participant, and then the average of the 5 runs was used for all further analysis. To analyze the quantitative data, we performed a repeated measures 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 [31] with six comparisons at $\alpha = 0.05$ was used.

*Section 5.3.4.1: Overall*

Average Total Time per technique is shown in Figure 3. We found significant differences between the techniques ($F_{3,27} = 76.2$, $p < 0.001$). With individual t-tests, we found that Raycast was significantly slower than Hook ($t_{26} = 9.665$, $p < 0.0083$), Scope ($t_{26} = 9.412$, $p < 0.01$), and Bendcast ($t_{26} = 8.796$, $p < 0.0125$). There were no significant differences in time between Bendcast, Scope, and Hook.

Average errors per technique are shown in Figure 4. Significant differences were found ($F_{3,27}$ = 269.1, p < 0.001), and further t-tests were performed. Raycast had more errors than Hook ($t_{26}$ = 18.189, p < 0.0083), Bendcast ($t_{26}$ = 16.657, p < 0.01), and Scope ($t_{26}$ = 16.543, p < 0.0125). Scope had more errors than Hook ($t_{26}$ = 3.737, p < 0.01667), and Bendcast had more errors than Hook ($t_{26}$ = 2.876, p < 0.025). There was no significant difference in errors between Bendcast and Scope.

Based on the post-hoc analysis, for the majority of scenarios, Raycast was significantly slower than the other three techniques. Scenario 7 showed significance in timing, but a Bonferroni adjustment eliminated any significance in the t-tests. For all nine scenarios, Raycast experienced significantly more errors than all of the other techniques. Only under two conditions did Hook experience significantly fewer errors than another techinque. For all other cases, there were no differences, which actually corelates well to how the participants rated the four techniques for accuracy.

Table 4: Completion Time Analysis

| | |
|---|---|
| Scenario 1 | $F_{3,27} = 8.69$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 4.054$, p < 0.0083 |
| Raycast > Scope | $t_{26} = 3.225$, p < 0.01 |
| Scenario 2 | $F_{3,27} = 17.5$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 5.711$, p < 0.0083 |
| Raycast > Scope | $t_{26} = 4.539$, p < 0.01 |
| Raycast > Hook | $t_{26} = 4.366$, p < 0.0125 |
| Scenario 3 | $F_{3,27} = 43.8$, p < 0.001 |
| Raycast > Scope | $t_{26} = 7.025$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 6.875$, p < 0.01 |
| Raycast > Hook | $t_{26} = 6.629$, p < 0.0125 |
| Scenario 4 | $F_{3,27} = 7.48$, p < 0.001 |
| Raycast > Scope | $t_{26} = 3.349$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 3.286$, p < 0.01 |
| Scenario 5 | $F_{3,27} = 21.9$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 5.11$, p < 0.0083 |
| Raycast > Hook | $t_{26} = 4.815$, p < 0.01 |
| Raycast > Scope | $t_{26} = 4.791$, p < 0.0125 |
| Scenario 6 | $F_{3,27} = 43.1$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 6.93$, p < 0.0083 |
| Scope > Hook | $t_{26} = 6.72$, p < 0.01 |
| Raycast > Scope | $t_{26} = 6.632$, p < 0.0125 |
| Scenario 7 | $F_{3,27} = 2.84$, p < 0.05 |
| Scenario 8 | $F_{3,27} = 6.85$, p < 0.001 |
| Raycast > Scope | $t_{26} = 3.133$, p < 0.0083 |
| Scenario 9 | $F_{3,27} = 33.1$, p < 0.001 |
| Raycast > Scope | $t_{26} = 6.87$, p < 0.0083 |
| Raycast > Hook | $t_{26} = 6.439$, p < 0.01 |
| Raycast > Bendcast | $t_{26} = 5.947$, p <0.0125 |

Table 5: Analysis of Errors

| Scenario 1 | $F_{3,27} = 41.7$, p < 0.001 |
|---|---|
| Raycast > Hook | $t_{26} = 7.366$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 6.755$, p < 0.01 |
| Raycast > Scope | $t_{26} = 6.447$, p < 0.0125 |
| Scenario 2 | $F_{3,27} = 38.7$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 6.827$, p < 0.0083 |
| Raycast > Hook | $t_{26} = 6.268$, p < 0.01 |
| Raycast > Scope | $t_{26} = 6.238$, p < 0.0125 |
| Scenario 3 | $F_{3,27} = 80.6$, p < 0.001 |
| Raycast > Hook | $t_{26} = 9.872$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 8.909$, p < 0.01 |
| Raycast > Scope | $t_{26} = 8.898$, p < 0.0125 |
| Scenario 4 | $F_{3,27} = 23.8$, p < 0.001 |
| Raycast > Bendcast | $t_{26} = 5.745$, p < 0.0083 |
| Raycast > Scope | $t_{26} = 5.300$, p < 0.01 |
| Raycast > Hook | $t_{26} = 4.648$, p < 0.0125 |
| Scenario 5 | $F_{3,27} = 34.0$, p < 0.001 |
| Raycast > Hook | $t_{26} = 6.730$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 6.056$, p < 0.01 |
| Raycast > Scope | $t_{26} = 5.419$, p < 0.0125 |
| Scope > Hook | $t_{26} = 3.362$, p < 0.01667 |
| Scenario 6 | $F_{3,27} = 64.4$, p < 0.001 |
| Raycast > Hook | $t_{26} = 8.393$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 8.182$, p < 0.01 |
| Raycast > Scope | $t_{26} = 8.142$, p < 0.0125 |
| Scenario 7 | $F_{3,27} = 13.8$, p < 0.001 |
| Raycast > Hook | $t_{26} = 4.276$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 4.200$, p < 0.01 |
| Raycast > Scope | $t_{26} = 3.883$, p < 0.0125 |
| Scenario 8 | $F_{3,27} = 34.8$, p < 0.001 |
| Raycast > Hook | $t_{26} = 7.018$, p < 0.0083 |
| Raycast > Bendcast | $t_{26} = 6.326$, p < 0.01 |
| Raycast > Scope | $t_{26} = 5.655$, p < 0.0125 |
| Scenario 9 | $F_{3,27} = 45.5$, p < 0.001 |
| Raycast > Hook | $t_{26} = 8.154$, p < 0.0083 |
| Raycast > Scope | $t_{26} = 6.799$, p < 0.01 |
| Raycast > Bendcast | $t_{26} = 5.980$, p < 0.0125 |
| Bendcast > Hook | $t_{26} = 3.059$, p < 0.01667 |

Table 6: Average completion time for (R)aycast, (B)endcast, (S)cope, and (H)ook

| Density, Speed | R | B | S | H |
|---|---|---|---|---|
| Low, Low | 3.26 | 2.01 | 2.22 | 2.43 |
| Low, Med | 3.89 | 1.98 | 2.28 | 2.46 |
| Low, High | 8.18 | 2.55 | 2.79 | 2.79 |
| Med, Low | 3.95 | 2.35 | 2.43 | 2.88 |
| Med, Med | 5.81 | 2.81 | 2.64 | 2.64 |
| Med, High | 10.4 | 3.26 | 3.56 | 3.36 |
| High, Low | 4.27 | 2.94 | 3.10 | 3.40 |
| High, Med | 6.91 | 2.94 | 3.93 | 3.41 |
| High, High | 9.09 | 4.16 | 3.85 | 4.05 |
| *Average* | 6.20 | 2.78 | 2.98 | 3.05 |

Table 7: Average errors for (R)aycast, (B)endcast, (S)cope, and (H)ook

| Density, Speed | R | B | S | H |
|---|---|---|---|---|
| Low, Low | 0.84 | 0.07 | 0.15 | 0.09 |
| Low, Med | 1.32 | 0.09 | 0.16 | 0.19 |
| Low, High | 3.22 | 0.33 | 0.31 | 0.17 |
| Med, Low | 0.90 | 0.16 | 0.14 | 0.24 |
| Med, Med | 1.66 | 0.35 | 0.44 | 0.22 |
| Med, High | 3.40 | 0.34 | 0.47 | 0.33 |
| High, Low | 1.12 | 0.39 | 0.41 | 0.34 |
| High, Med | 2.25 | 0.53 | 0.53 | 0.32 |
| High, High | 3.12 | 0.84 | 0.65 | 0.39 |
| *Average* | 1.98 | 0.34 | 0.36 | 0.25 |
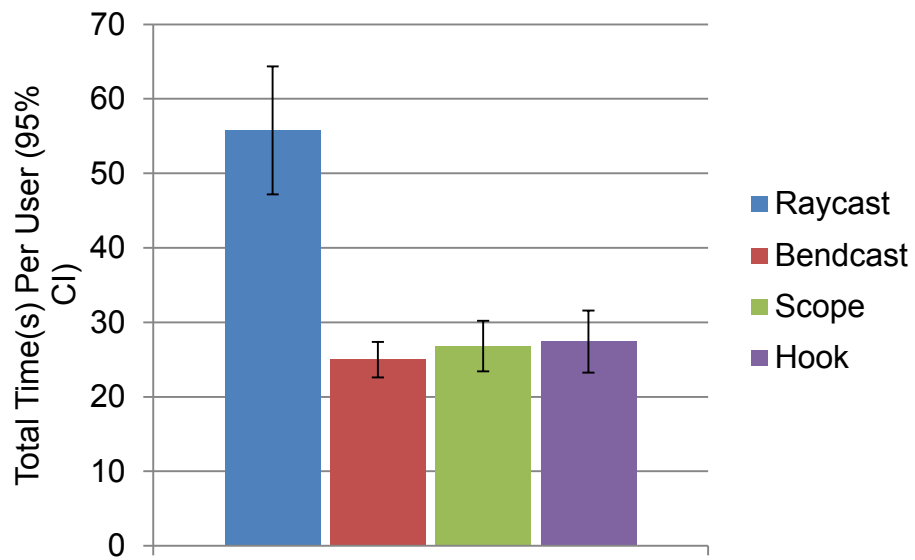
Figure 5.1: Mean Total Time, All Scenarios
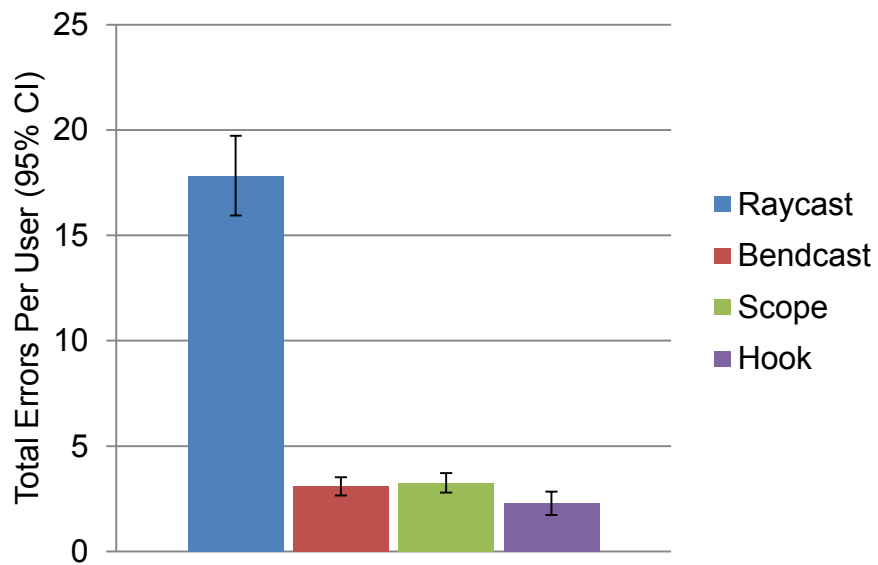


Figure 5.2: Mean Total Errors, All Scenarios

*Section 5.3.4.2: Post-Questionnaire*

The Post-Questionnaire used a 7-point Likert scale for all of the questions except for question 6, which asked the participants to rank the four techniques from 1 (most preferred) to 4 (least preferred). Questions 2, 3, 4, and 5 asked the participants to rate the usability, speed, and accuracy of Raycast, Bendcast, Scope, and Hook, respectively (see Figure 5). We performed a Friedman test on Q2-5, and determined that there was significance. We then performed a Wilcoxon Signed Ranks test to determine the significance. Raycast was significantly lower rated for usability and speed than the other three techniques with a significance p-score of $< 0.001$. The Z-values ranged from -3.85 to -4.309. Raycast was also rated significantly less accurate than Bendcast (Z = -2.678, p $<$ 0.0083), but not significantly less accurate when compared to Scope or Hook. Between Bendcast, Scope, and Hook, there was no significance for any of the three attributes.

Figure 6 shows the participant rankings for most and least preferred technique, asked in question 6. We performed a Chi-squared test on this question to determine if there was a preference for any one of the techniques. An analysis on the most preferred technique shows that Scope was preferred over the other techniques ($\chi_3^2$ (N=27) = 9, p $<$ 0.05).

Additionally, Raycast was the least preferred technique ($\chi_3^2$ (N=27) = 41, p $<$ 0.01).
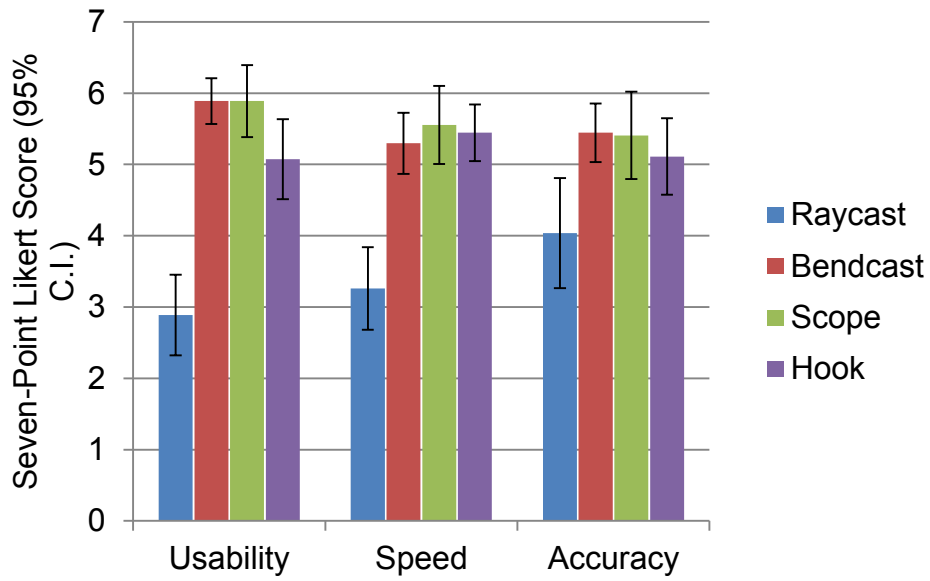
Figure 5.3: Post-Questionnaire, Technique Critique
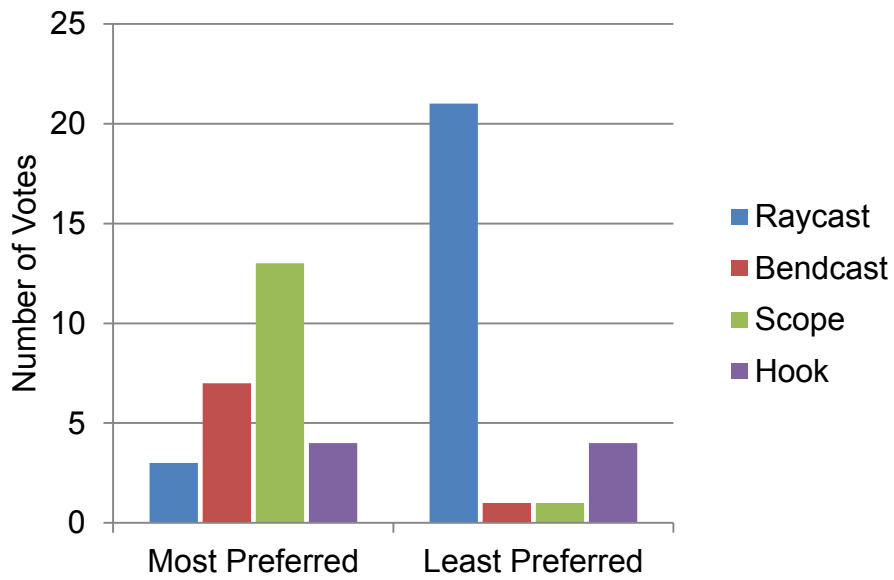


Figure 5.4: Overall Ranking of Selection Techniques

Section 5.4: Discussion

Out of the four techniques, Raycast was the only one that did not feature any sort of aid or assistance to the user. The extreme level of precision in the selection process is a major contributor to the increase in errors, and thus completion time. In six of the nine scenarios, it had significantly more errors than each of the other techniques. The only scenario which exhibited no significant difference in completion time between any two techniques was scenario 7 (high density, low speed). In a high density situation, there is likely going to be a lot of occlusion. This behavior was observed during the study while monitoring participants. Regardless of which technique is in use, the occlusion can add considerable time in finding and selecting the target object. This burden is what likely evened the playing field between all of the techniques. In this scenario, Raycast did experience significantly more errors than the other techniques, but it still only averaged 1.1, which is relatively low when other scenarios are considered. In fact, Raycast experienced significantly more errors than all of the other techniques for every scenario. This explains why it appears so error-prone in Figure 4.

Between the remaining techniques, there was no significant difference in completion time when considering the average total time for each technique, per user. These techniques all provided some form of aiming aid in the form of selecting the object that is closest to the center of the cursor. Additionally, there wasn't a single scenario that featured a difference in completion time. There were three instances were significance in completion time was close, but was ruled out due to a Bonferroni adjustment. We believe that in scenarios where cursor velocity would stay relatively high, that Scope

would show an advantage over Bendcast due to its increase in selection area. It is possible that users were not comfortable with the idea of selecting an object that is at the edge of the cursor, but instead tried to center it first, thus negating the increase in operational flexibility. Between Scope and Hook, they do both perform a similar historical analysis of object distance to the center of the cursor and compute a winning object per frame, so their performance was expected to be similar. The perceived advantage that we expected Scope to have with its changing visual appearance ended up not providing any performance increase, but did show that it was an obvious favorite with the users. The instant feedback that the users received likely contributed positively to their confidence in using the technique, and thus more comfortable using it. If additional visual feedback could be incorporated into Hook, then perhaps users would appreciate using it more.

Hook did experience fewer errors than both Bendcast and Scope overall. The design of Hook utilized a crosshair type cursor, which was very similar to the one used for Raycast. Because of this, we suspect that users were likely to be cautious when selecting, since they would have had a difficult time determining if they were using Raycast or Hook at the moment. This could explain the reduction of errors, and the slightly higher (although not significant) average completion time. Out of the nine scenarios, Hook experienced significantly fewer errors than Scope just once in scenario 5 (medium density, medium speed) and significantly fewer errors than Bendcast just once in scenario 9 (high density, high speed). Without downplaying this statistical

significance, we do suspect that these differences are possibly an anomaly, since they were so isolated.

## Section 5.5: Summary

We have taken ideas from existing techniques and molded them into a single dynamic technique that adapts its behavior based on user input. Our user study compared its performance to that of several existing techniques, and revealed that although it performed similarly to the other non-Raycast techniques, it was strongly preferred over all of the other techniques. We also further showed that more modern techniques have pulled ahead of Raycast, so much so as to render its use all but unnecessary. Our initial research has shown that there is potential to further explore the area of dynamic selection techniques. When designing a selection technique that must adapt to both user behavior and the environment, there are many aspects to take into consideration. This topic is the focus of the next chapter.

# CHAPTER 6: ADAPTABILITY OF NOVEL TECHNIQUES

## Section 6.1: Introduction

The design of existing selection techniques vary greatly. The manner in which they were designed or programmed can be done in a manner that allows for tuning of certain attributes to ensure optimal performance either in a domain-specific area, or in a broad range of selection scenarios. Techniques can be designed to work best with a specific form of input, such as touch, mouse, in-air controller, etc. Sometimes these techniques work across multiple input types. Sometimes they only work with a specific in method in mind.

The act of selection is one of the primary functions of any human-computer interface [2]. Given this, it is no surprise then that selection techniques have been studied quite extensively. Stemming from the natural method of pointing in real life, the point cursor was and continues to be the basis for virtually all future selection techniques. It exists in many forms, both as a single point in a 2D plane as in WIMP interfaces and as a casted ray in 3D environments [62]. As more complex techniques were developed from this, taxonomies inevitably followed, just as in other areas of research [48] [9] [2]. These taxonomies aided in developing a greater understanding of how these techniques worked, and highlighted the overall development space that had been explored over the history of selection research. These taxonomies are all similar in that they study the static nature of each technique, making comparisons to how they operate at a high

level. What is missing from this body of research is a more detailed study of how the design of the techniques could be better adapted to a broader use case.

In our research, we propose a more detailed, in depth analysis of existing techniques, and how they can be better adapted to a broader range of selection environments. We also take special considerations to how each technique can be adapted to integrate into an intelligent selection framework that permits the dynamic switching between techniques, known as Auto-Select (Figure 4.1) [12]. The design of each technique covered here is scrutinized and considered for what specific attributes play the most significant role in its adaptive behavior. By performing this analysis, we can come to a deeper understanding of how to make existing techniques more versatile. To aid in a more qualitative discussion, topics associated with establishing a more valuable taxonomy are illustrated.

## Section 6.2: Considerations for Dynamic Selection

The motivation for performing a detailed analysis of dynamic techniques stems from the lack of consideration for the dynamic nature of how they were designed to operate. We are not simply interested in the more obvious aspects, such as control methods or disambiguation methods, but more so in how these techniques were designed, and what features make them unique from pre-existing techniques. In addition, we also consider attributes that aid in the inclusion of a technique in the Auto-Select framework. Because of this, we will consider several less discussed aspects of the techniques, and how that impacts the usability of the technique in a broader range of use cases.

*Section 6.2.1: Design Motivations*

When evaluating a selection technique, it is important to review why the technique was created. This can aide us in properly classifying it with other techniques that are designed for the same overall purpose. Generally speaking, there are three different reasons why a new technique is created. The first is the most obvious: improving on a previous technique, given the same set of circumstances and use cases. This describes the more common evolutionary design approach, and is quite valuable. The second reason is due to some innovation in hardware that allows for novel techniques based around it to be designed. A great example of this is the Go-Go Interaction technique, that was designed for the 6-dof sensors that allow for physical hand tracking and mapping to the virtual world [47]. The third reason for technique creation is the design of new environments where a technique could be applied. This is often driven by the advancement of processing power, allowing for new and more detailed selection scenarios that need more optimized techniques. Another implication of processing advancements is the leveraging of this power to enhance the algorithmic or visual component of a technique, making it either more accurate or more relatable to the user.

*Section 6.2.2: Interaction Levels*

When considering which techniques could be grouped together into an Auto-Select framework, the interaction level needs to be the same. By this, we mean that they are all understood to use the same type of human-computer mapping [input device], and

operate in the same 2D/3D interactive space in the interface. This will help to minimize the disruption to the user during the technique switching. These interaction spaces can be broken into three different types: Planer 2D, 2D-overlay, and Interactive 3D.

Planer 2D are techniques that exist in a 2D plane that do not have any 3-dimensional component or knowledge. They are almost exclusively found within 2D interfaces, such as the typical cursor in WIMP interface. Sometimes, 2D interfaces that compliment 3D interfaces will use these types of techniques to permit selection that is not 3D in nature, like a settings menu to a game or simulation.

Image plane techniques are those that are rendered in a 2D plane that overlays a 3D scene, but are able to interact and utilize 3D data [9]. This usually involves a 2D visual component to the technique, with operation that involves 3D knowledge to make a selection decision. This is the space where a large portion of techniques operate [10] [35] [45].

Interactive 3D techniques are those that are rendered within the 3D scene, such as a virtual hand [47]. An effective model of this is the flexible pointer, which allows the user to use one hand to control the general direction of a 3D ray that exists in the scene, and use the other hand to dictate how the final portion of the ray should bend to aid in selecting obscured objects [44].

*Section 6.2.3: Speed Versus Accuracy*

There are scenarios where accuracy is favored over speed, and vice versa. In some cases, a user's completion time has an effect on the quantitative measurement of success. Examples of this might include shooting enemies, or slicing fruit. These are typically action game realms where the user is in a live real-time environment, with things happening all around them. Timing plays a critical role, and the user is encouraged, if not explicitly, to perform quickly at the expense of some accuracy. This tradeoff that the user makes is expected to be less costly as their skill level increases, as with most things a user gets more accurate as they increase in skill at performing a task.

*Section 6.2.4: Impact of Error*

The impact that an error has on a task will directly affect how a user balances their speed versus accuracy. An error can cause one of three types of penalty: explicit, implicit, or none. An explicit penalty is one handed down by the software itself, such as a depletion of ammunition, damage done to the player, etc. Something that the player had is now lost, or the users experience is directly negatively affected. In cases where this type of penalty does not exist, but time is lost or spent making an incorrect selection when time is of importance, then this loss of time is an implicit penalty, a sort of opportunity cost. This type of scenario can lead users to attempt to game the system, often by performing rapid selection [if possible] without care for accuracy, since no

explicit penalty is awarded. In scenarios where time is not being measured and no explicit penalty is awarded for an error, then there is no penalty at all.

The gradual improvement of selection techniques should have a positive impact on the occurrence of user error. In a situation where techniques are blended using the Auto-Select framework, the error penalty should be roughly equal for all techniques. If a technique is a multi-step one such as SQUAD, then there is some level of penalty with the user required to identify that the target is not visible beyond the first step, and now they must back out and undo their selection.

Section 6.3: Selection Technique Analysis

*Section 6.3.1: Point Cursor*

The Point Cursor is a fundamental method of selection that is analogous to pointing at something with a finger and touching it. It is the most direct method of selecting something, requiring very minimal processing and design overhead. In its most basic form, it is represented by a cursor icon which the user controls by moving an input device. Selection is triggered by a physical event such as a button press. The cursor icon features a small area, often a single pixel or point that determines which object will be selected. In a modern desktop operating system, the icon is usually an arrow, with the selection point located at the tip of the arrow. In many other systems, the icon is a cross-hair of sorts, with the selection point located in the center of the cursor.

Raycast is a 3D variant of Point Cursor that usually only differs by the fact that the selection point functions by projecting a ray into the 3D scene to collide with a potential target. In some cases, the ray can be rotated by using a high-DOF input device, which allows the user to point around objects.

*Section 6.3.1.1: Attributes*

Although basic, there are several attributes that can be analyzed for modification.

*Cursor Size*

The size of the cursor is important because it directly impact how well the user is able to locate and track it while in motion. The size can also play a role in relaying information about the selection conditions.

*Cursor Image*

The image used for the cursor is very opportune as a tool for informing the user, more so than its size. In most all modern graphical interfaces, the cursor image changes to suit the object being interacted with, such as a button, hyperlink, text box. These are examples of feedback that occur before the selection event, to educate the user as to the possible outcome a priori. Changing the cursor image in flight is another viable way of reducing selection time [21]. A more generic form of feedback by means of cursor image manipulation is to indicate that the user placed the cursors activation point on top of a valid (or invalid) target.

117

*CD Ratio*

The CD ratio is the relationship between the movement of the input device and the display representation, e.g. the cursor. When using a standard computer mouse, this has the displacement of the input match the displacement of the cursor. In a joystick setup, the displacement of the joystick is mapped to the velocity of the cursor. The ratio is not always linear, as represented by the acceleration feature, which has the primary goal of increasing low-speed accuracy without sacrificing usability overall.

*Section 6.3.1.2: Analysis*

The most basic form of these techniques would use a fixed CD ratio. In that case, there is no specific adaptation for dense or dynamic scenarios. By modifying the CD ratio based on user input speed, a form of acceleration can be achieved. In a 2D physical space such as a desktop, this does not have any serious side effects, but in a 3D tracking world, acceleration would cause an explicit drift in the mapping from physical location to virtual location, which would hinder the experience. In any case where user input is performed by pointing a device at the screen, the concept of acceleration is eliminated.

Since this technique uses a single point and not an area for determining potential targets, there is little need for disambiguation. However, it still leaves open the possibility of choosing between targets nearest the activation point. In 3D form, this can be seen in Snap-To-Ray (Figure 6.1) [62]. In 2D form, this was accomplished by

Bendcast [12], among others. When this snapping is being performed, more variables open up to being modified dynamically. The maximum distance at which an object will be snapped to is tunable, as well as the manner in which the user is presented with the knowledge of the snap taking place. In the 2D interaction level, the snapping could be visualized by an auxiliary arrow extending from the cursor over to the snapped object, or by a second phantom cursor that resembles the original, but different in some distinct way, such as its color or transparency.

*Section 6.3.1.3: Auto-Select Considerations*

This technique serves as the foundation for virtually all other techniques. This is analogous to something of an "Identity Technique", lacking in all but the most basic features required to perform a selection. Being that this is the case, it can serve as a fallback technique when other, more specialized techniques are no suitable. With any technique that features a relatively conservative cursor, it could fall back to point cursor when conditions are right, such as with slow moving objects of decent size that are easy to target without any aids. Some techniques such as Expand [10] act just like a point cursor when only one object exists in the cursor. This is a way of reducing a two-step technique down to a single step when the need arises.
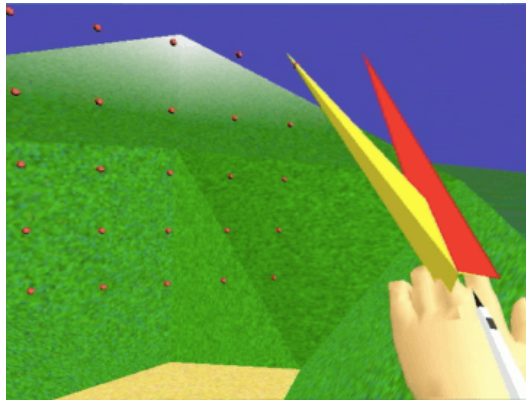
Figure 6.1: Snap-To-Ray: Red (right) is user pointing position, yellow (left) is snapped ray snapping to nearest object.

*Section 6.3.2: Aperture*

Aperture is a technique designed for 3D immersive environments [24]. At its core, it is an area cursor, based off of the spotlight technique [38]. To control the size (aperture) of the area, the user moves a physical ring either closer to or farther from their dominant eye. The ring is attached to a wand that features 6-DOF tracking, so that the software can accurately compute the effective area visible within the ring by the user (Figure 6.2). This means that the user controls the position and size of the cursor with the same input device, and is able to move it in any direction in physical space, seeing a change in their virtual selection area and position.

*Section 6.3.2.1: Attributes*

The primary differentiator of this technique is the method that the user controls the area of selection. Otherwise, the nature of the technique otherwise is similar to many other techniques. As a result, the defining feature of this technique will be considered below.

*Aperture Size and Shape*

At its very core, the size of the aperture is user-adjustable. In some sense, the technique is dynamic, but it is actually dependent on the user's own input, not any scene-aware intelligence. The physical component is the circle and wand. If a physical circle is used, then it is not something that could be changed, but if a transparent display is used, then the aperture circle itself could be a dynamic component that changes based on conditions. It could change shape, size, color, or translucency to aid the user in achieving optimal selection.

*Section 6.3.2.2: Analysis*

While it operates as a stand-alone technique, it is primarily a contribution to physical methods of controlling selection. As described above, there are numerous ways that the physical aperture could be modified. This leads to a potentially very interesting combination of dynamic semi-physical content, in addition to all of the content presented on the system display.

There are several challenges with the technique that make it more difficult to adopt. The user must indicate their dominant eye to use as the source point. This will always cause a double image to appear to the user because they will see the wand with the non-dominant eye, possibly leading to a distraction. Also, the user is likely focusing on the screen beyond the aperture circle, causing the circle and wand to go out of focus. The blurriness can interfere with the user's ability to discern objects. In addition to the visual distractions, the user experienced fatigue in the arm holding the device after a short amount of time. This was lessened by the use of a longer wand, allowing the user to lower the arm to a more comfortable position. Despite this solution, there will still be some level of user discomfort after extended periods of time.

*Section 6.3.2.3: Auto-Select Considerations*

For this technique to be woven together with other selection techniques, they must also utilize the wand to capture user input for manipulation. From this, it could be imagined that a modification to how the volume changes its shape would be the primary means of switching selection behavior. This technique could be utilized alongside a point cursor, which is just the spread radius reduced to zero, in its basic form.

*Section 6.3.2.4: Multi-Step Considerations*

As is, aperture is a single step selection technique. To aid in disambiguation, it could be modified to perform multiple steps of selection. Optionally, it could operate in different modes.

One such mode could be a zoom function that takes what is within the aperture and magnify it, much like a magnifying lens. This would work well when there are a few objects within the area that could be enlarged to provide the opportunity to exclude all but one within the aperture.

By taking advantage of the shape of the ring, it allows for potential targets to be displayed around it. Selection in step one leads to a second step that provides the user with an opportunity to select from a reduced list of non-interfering objects.

A combination of these two ideas would provide a method that not only increases the size of the objects, but also spaces them out in a logical manner, allowing for very simple and effective selection (Figure 6.3). In complex scenarios, an algorithmic approach could be taken in order to cap the number of targets advanced to the second round, such as picking those which are closest to the center of the aperture.
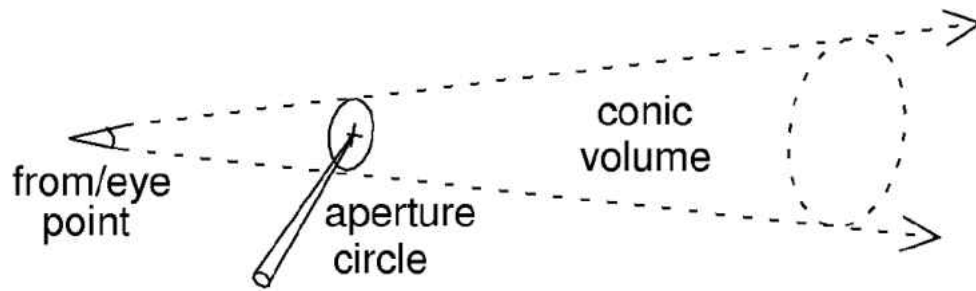
Figure 6.2: Aperture Selection Method
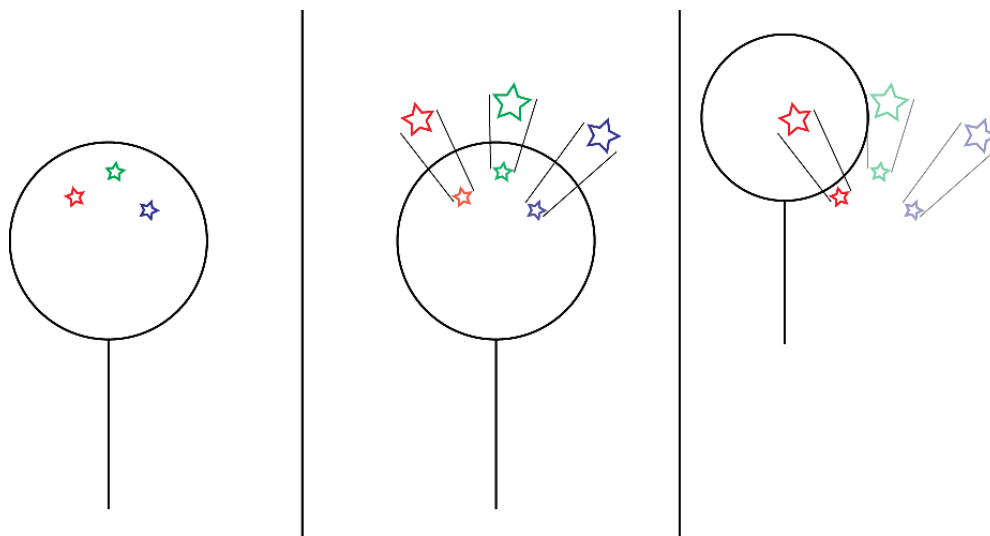


Figure 6.3: Multi-step disambiguation. User hovers over potential targets (left). User performs selection, triggering disambiguation step, showing objects in a pseudo-circular fashion (middle). User moves input device up and to the left, thus eliminating the ambiguity and makes a clear selection (right).

## Section 6.3.3: Bubble Cursor

Bubble cursor was designed for 2D interfaces, and operates in the Planer 2D interaction level [26]. It is an area cursor that features a dynamic area size, dependent on the proximity of the nearest target. The area will adjust to the size needed to encompass just the nearest target, thus eliminating the possibility of ambiguity (Figure 6.4). The inspiration for the design was the desire to apply Fitts' Law [39] to create a technique that minimized the negative aspects and maximize the positive ones. This equates to decreasing the distance to the target, while increasing the effective size of the target. They achieved both to some extent by increasing the size of the area, bringing it closer to the target. This technique was shown to perform better and be more preferred versus techniques that feature dynamic selection areas [29].

Two derivations of bubble cursor were designed and tested by Laukkanen et al [37]. Lazy bubble cursor differs in the way it grows the bubble. It is done more slowly, making the feature less distracting to the user. It also allowed for the selection of empty space, a feature not present in the original bubble cursor. The growth function was designed to only operate if the nearest target is no more than half the distance to the cursor as the second nearest target. The second derivation was the cone cursor, which rendered a tear-drop like shape that stuck to the previously selected object. This was done with the goal of reducing the overshooting of targets. In their results, they found that the lazy bubble had significantly more errors than the standard bubble. This can be explained by the fact that it was possible to select nothing, which can only lead to an increased number of errors.

### Section 6.3.3.1: Attributes

Given that it is already a dynamic technique, there are several attributes to consider, some of which have already been experimented with as described above.

*Area Size*

The size of the area is presently tied to the proximity of the nearest target. This is a fundamental aspect of the technique, intended to always be large enough to encompass the nearest object.

*Area Growth Rate*

The growth rate is something that can be modeled after a variety of mathematical functions. This can be tied to a variety of things, such as distance to object, cursor velocity, or user preference, to name a few.

*Area Shape*

Originally a circle, it was proposed that the shape could be elliptical [26]. It has since been modified to look like a cone, sticking to targets. Other dynamic shapes may still help to improve selection performance.

### Section 6.3.3.2: Analysis

The technique was designed and tested in static environments with no motion. The introduction of motion could cause some interesting side effects that might make it

harder to select a target. If the cursor is chasing a moving target, any objects in the path traveled could attract the bubble, making it quite difficult to actually select the target until it is in a more sparse location. If an over-time scoring algorithm was used, this problem could be mitigated. Those objects that are briefly passed over would not have a significant score over time, thus causing the bubble to envelop the intended target sooner and with less resistance.

It was identified that the performance enhancement of the bubble was valid down to the point where the effective width was as little as 33% greater than the actual width of the object. This shows that there is a point where the technique breaks down, and a more suitable alternative might be desired. This is not a fault of the technique, but merely a natural limitation of the method used.

Adapting the shape of the area has potential, especially when combined with cursor movement. It is a possibility that taking the central cursor position and combining it with the direction of movement to create a new virtual positon from which the area is computed can cause objects that lead the cursor to be more likely to be enveloped. The shape of the area could resemble that of the cone of uncertainty, used to project the possible future position of a hurricane, much like that used in the Implicit Fan Cursor [56].

Since this is a 2D technique, it would integrate well with many other 2D techniques. In high density scenarios, it could switch to a multi-step technique such as Expand or SQUAD for improved object disambiguation when hovering over densely packed stationary objects. When moving beyond a certain rate of speed, the cursor could switch to an adaptive Implicit Fan Cursor that can optionally preserve the target-enveloping function for improved target selection. In cases where objects are sparse and the cursor is relatively far to the nearest object, it might be beneficial to inhibit its enveloping function, since that could cause unexpected behavior and confuse the user. Since bubble cursor was already shown to out-perform a point cursor in virtually all scenarios [26], it is unlikely that it would ever stand to benefit by switching to it.
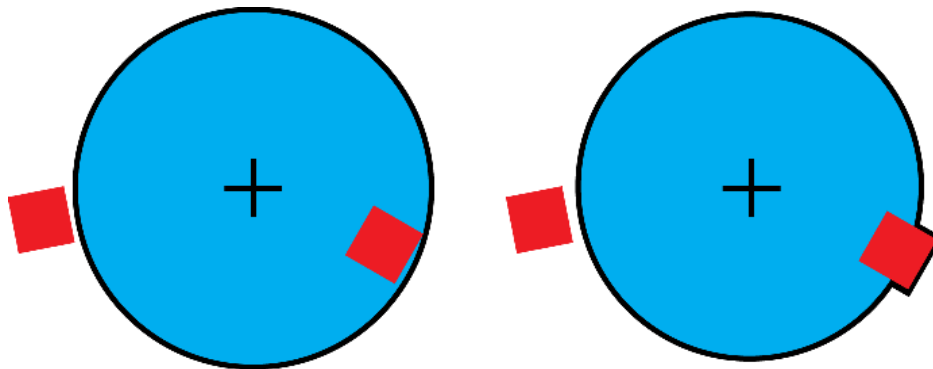


Figure 6.4: Standard area cursor (left) encompassing a single object. Bubble cursor dynamically sizing to encompass single object, including outline feedback (right).

*Section 6.3.4: Scope*

Scope was designed as a dynamic technique that bridges the gap between a point cursor and an area cursor [11]. This is achieved by dynamically scaling the cursor area as a function of its velocity, based on the understanding that a cursor in motion is not only harder to visually track, but also harder to select with [10]. It was originally based on DynaSpot [14], which also features a dynamically sized area cursor. Where Scope differs is the inclusion of a scoring algorithm that acts as a disambiguation method, which runs over time to better anticipate the desired object to select. Another distinguishing feature is the way in which the cursor looks. In contrast to the shaded circle of DynaSpot, Scope is represented by a circular ring that provides more space for the user to view what is inside. Another difference is that DynaSpot will reduce to a single point (pixel) when still, while Scope remains as an area cursor. This last difference was done as a response to testing, which showed that performance was virtually always better with an area versus a point.

DynaSpot, which Scope was based on, was created for and tested in a planer 2D interaction level. Scope took this to another level by operating in a 2D overlay on a 3D scene. Information about objects was gathered by raycasting into the scene.

Several dynamic characteristics of both Scope and DynaSpot will be highlighted here.

*Cursor Size*

The size of the cursor is an attribute that is already leveraged to make a dynamic technique. The limits to how large or small it is permitted are also an important detail. This is one of the ways that these two techniques differ.

*Growth Rate*

The observable rate of growth for both of these techniques is quite similar. They each feature an initial delay and ramp up period, followed eventually by a reduction period. Just as is the case with tweening animations, the easing of this rate can be adjusted on the fly if it improves selection.

*Disambiguation Method*

Scope uses an over-time method of improving the accuracy of guessing which object the user intends to select. This can be done in a variety of ways. Such aspects include which objects to keep score on, how to score an object based on distance, how much emphasis to put on scores from past measurements, and how long to retain data for scoring. Scope measures over 0.5 seconds, counts objects only within the area of the cursor, and uses an exponential moving average with each sample being ½ as significant as the one more recent.

Both Scope and DynaSpot showed considerable improvements in performance over raycasting [11] [14]. They both dynamically adjust the cursor area size in relation to speed, but Scope takes it a step further and performs some disambiguation over time to improve selection accuracy. The other differences between the two techniques are minor, and can be considered minor visual tweaks.

The level of dynamic adaptation was limited though to a single dimension: cursor velocity. Since this did show that it contributed to improved performance, perhaps there is another way that the technique could adapt. With bubble cursor, the area was adjusted based on the distance to the nearest target. If something similar was implemented, then Scope could be improved. At low speeds, the cursor is small. A smaller cursor will increase the difficulty of capturing a target, but also reduce the likelihood of multiple targets inside of it. This might be the perfect opportunity to integrate a bubble cursor type of adaptive sizing that will adjust the cursor size, within certain limits, to only select a single target.

Another method of adjusting the area cursor to improve low-speed selection in dense conditions would be to offset the cursor, instead of resize it. The way this could work is when the cursor is below a size threshold, it is eligible for offsetting. The nearest object would be measured, and if the cursor is within the range of it, then it will offset to encompass just that nearest object (Figure 6.5). This could be an animated offset adjustment, with possibly some distinctive visual feedback to better inform the user as

to this function. In our example, we adjusted the color of the cursor to indicate this was occurring.

*Section 6.3.4.3: Auto-Select Considerations*

The strength of this technique is its ability to operate well in sparse to moderately dense environments. In cases of high density, the same problem of accuracy versus ambiguity occurs. To address this, a multi-step disambiguation method would be helpful. In this case, Expand or SQUAD would complement Scope where it is weakest. The visual indicators of the technique would not even be impacted, and the user would not experience any significant impact.
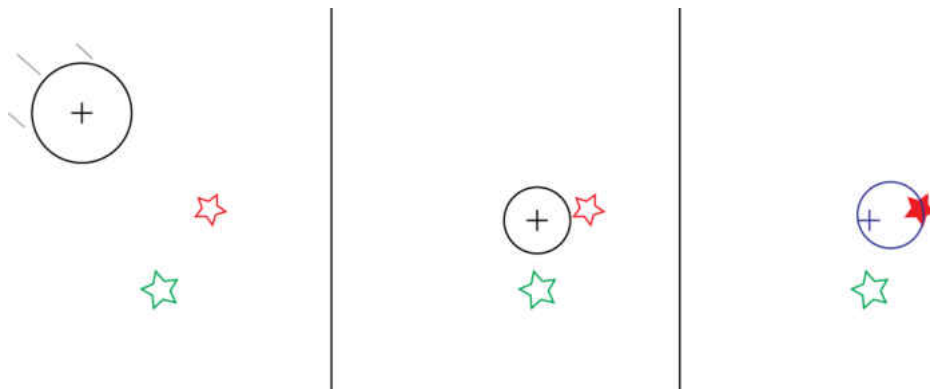


Figure 6.5: Scope Dynamic Offset. Cursor is moving down and right towards objects (left). User slows cursor to a near-stop to prevent overshooting target in red (center). Cursor dynamically performs area offset to encompass nearest target (right).

*Section 6.3.5: Implicit Fan Cursor*

The implicit fan cursor is a novel approach to selection that leverages user input to make a more intelligent decision of which object to predict as a target [56]. It works by generating a fan-like shape in the direction of the cursors movement, for all cursor velocities. The angle, or spread of the fan varies depending on the speed of the cursor movement. At its narrowest, it was set to 90 degrees, and at its highest speed, it would extend to a full 180 degrees. The nearest object that lies within the angular range at the time, regardless of distance, is chosen as the potential target, and indicated as such with a gray outline (Figure 6.6, left side). The basis for the improvement in performance over another technique like bubble cursor is the fact that objects in the direction of movement are considered, while objects which the cursor is moving away from are not considered.

In its original design, this fan was not only the activation area, but was actually rendered on the display. In preliminary testing though, it was discovered that displaying the fan resulted in the highest average movement time, and higher errors than just showing a crosshair. Interestingly, the circular cursor like what is used in many other area cursors had only slightly lower movement time than displaying the fan, but nearly a 40% higher error rate. The crosshair style had the lowest average movement time and the lowest error rate. In support of their quantitative data, their qualitative feedback contained several comments from users reporting that the visual feedback of the fan was distracting.

The uniqueness of the technique indicates several configurable attributes, of which are several are already dynamically adjusted but could be considered for additional modification.

*Visual Representation*

As was already tested in their second preliminary study, representing the fan as simply a crosshair cursor was shown to provide the best performance. Other options included something resembling the bubble cursor, and another that was the actual fan of the activation area, displayed for the user to see. There is an opportunity for trying other forms of representation, to see if any can outperform the crosshair.

*Fan Angle*

The opening angles were tested, and vary with speed. This includes how the angle functions when the cursor is motionless, or nearly so.

*Fan Depth*

The depth of the cursor is presently dictated by the distance to the nearest object that is within the projected path of the cursors opening angle. It is possible that other shapes besides partial circles could be employed to better focus on correct potential targets.

*Section 6.3.5.2: Analysis*

For the sake of experimentation, the ability to select empty space was omitted from the technique design. This raises the question of how the technique would best be modified to permit this function. As is, the technique stays in whatever the previous state it was in. This means that there will [typically] be a target highlighted for pending selection, with no way to unselect it. A simple way to solve this would be to utilize a physical method, such as pressing an alternate selection button. A similar way to do it in software would be to require a press-and-hold for some minimum amount of time to distinguish from a normal selection attempt. To solve this algorithmically though would take much more consideration.

Selecting empty space without any explicit special action from the user would first require that no object is selected. This requires that the technique have a way to release the last target, or exclude itself from always selecting something. This could be achieved by having a minimum distance required for any object to be within for the selection algorithm to activate. This would be a relatively simple modification to integrate that does not affect the primary characteristics of the technique. Another method, though much more difficult to integrate, would be to recognize a shake gesture to virtually "shake off" the selected target. This would obviously require explicit user action, and have the potential to trigger false positives during normal use if the recognizer isn't accurate enough.

This technique always simply chooses the closest target to the cursor, within the bounds of the spread fan. This opens up the possibility of an off-angle object being chosen when there is a more direct, potentially more likely object that could be chosen (Figure 6.6, right side). Taking this technique a step further, the actual angular offset from the true direction would likely be a beneficial modification.

This technique was only tested in static scenarios. With moving objects, the way that this technique interacts will be quite different. With moving objects, the user can often be chasing a target, all while flying by and over other non-relevant ones [10]. Without any form of disambiguation or scoring, the algorithm will falsely select all sorts of objects that it is flying by randomly. It is because of this that an over-time scoring algorithm should be integrated. A basic closest-object over time accumulating score utilizing an exponential moving average over the past ½ second or so would be a good start. This would allow the random objects to only accumulate a relatively small score when compared to objects that are being chased, leading to a more accurate and less frustrating selection experience.

The implicit aspect of the technique refers to the fact that the fan is only represented by the activation area, but not rendered on the display. The user is left with little visual feedback to be aware of which target is selected. With the absence of any maximum distance to consider, this permits the possibility of a rather distant object being chosen. It also leaves the cursor visually disconnected from this chosen object. For some, this might be confusing. An easy solution that we propose would be an arrow that connects the cursor crosshair to the chosen object. As the chosen object changes, the head of

136

the arrow can transition its position in a smooth manner to minimize any visual disruption and actually lead the user with a visual cue as to where their selection has changed to.

With a stationary cursor, there is no movement data to indicate which direction to point the fan, other than the movement last captured. Pointing the fan in a direction that the user is not moving the cursor in at the moment might leave them feeling disconnected from the intended action, stillness. Because of this, we suggest that the technique transition into an area cursor of some minimum radius when below some minimum speed. It seems understandable that at very low speeds, the user might want a regular circular area cursor to select something that is close to the cursor, without excluding entire regions due to subtle cursor movement. This will also aid in selecting an object that has been overshot, and the user is making a minor correction, such as in the cone cursor [37].

*Section 6.3.5.3: Auto-Select Considerations*

This technique is suited well for static environments, as well as those with moderate density levels. In situations where density is very high, there is still a need for more disambiguation. This would very well be achieved by a two-step such as Expand or SQUAD. In lieu of doing that, the technique could instead be modified to have this functionality built in, which would likely lead to a smoother transition of functionality. On the other extreme, in conditions of sparseness, the directional enhancement would not

provide much, and thus the technique could revert to a simpler, bubble-type cursor that simply looks at nearby objects without concern for cursor movement.
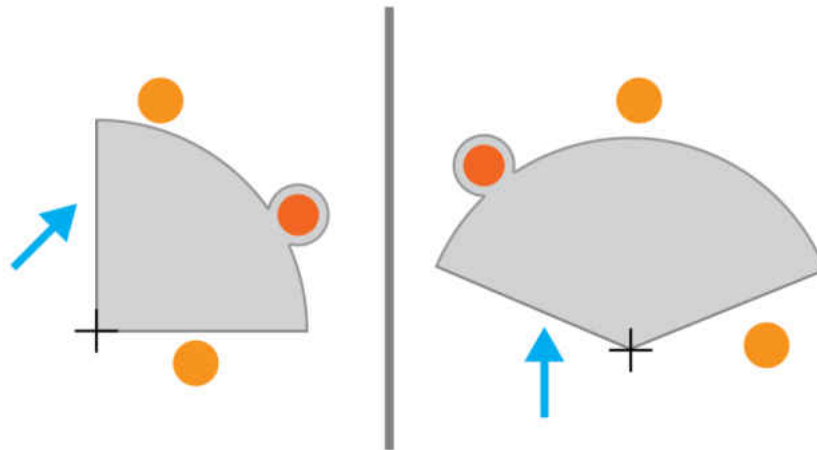


Figure 6.6: Implicit Fan Cursor. User selects object closest to the cursor that is within ran spread (left). Object off to the left is favored over the object directly ahead of the fan (right).

Section 6.4: Discussion

Our walk through the technique studied have taken us from the more basic to the more recent and advanced for all of the selection techniques reviewed, a suitable strategy for incorporating into the ASF was presented. We also were able to design some new methods for extending these selection techniques in ways that demand future work and studies. Aperture, the only technique studied that uses a physical analogy for selection (the wand), stands out from the others. Our idea of using secondary, transparent display to augment the selection process is an interesting proposition. By making the wand

shape (presently a circle) digital via a translucent display, it could be changed at any time to either suit the user, or to alter the selection process.

The Implicit Fan cursor features a lot of possibilities for future work. We discussed many ideas, and believe that by exploring these, a stronger technique could be created. We would also like to see how the technique works in 3D environments, as a 2D overlay. The added depth information of an object could affect how objects are considered for selection. Also, the notable lack of a scoring algorithm for improving selection accuracy is something we would like to see tested when included. The significant opportunity for including this and the other techniques into the Auto-Select framework is a task that we would like to perform and test, to validate our initial research findings.

## Section 6.5: Summary

The potential improvements to these selection techniques that we have proposed offer a way for them to work better not only as is, but in conditions that they were not targeted at. With the Implicit Fan Cursor, our suggestions for low speed control would offer better performance by making its behavior and appearance more intuitive and in line with user expectations. Our two-step selection idea for Aperture was designed to improve performance in high density scenarios, thus improving its performance envelope. These are both examples of how we answer the final research question regarding the improvement of selection techniques to increase their performance envelope.

The goal of this analysis was to bring to light the more recent development of dynamic selection techniques and to offer an initial study in to how they operate. By studying these techniques, we were able to gather information on how these techniques could not only be improved, but utilized in conjunction with the Auto-Select framework. This will make it possible to join together various techniques that work well in some cases, but not others, and maximize the use of each technique when it is most suitable for use. Our future work will include taking our discoveries from this research and applying them towards developing a new selection technique, one that addresses the shortcomings of those we studied in detail. A summary of our improvements is found in Table 8.

Table 8: Feature Summary for Selection Techniques

| Technique | Feature Ideas | Multi-Step | Auto-Select |
|---|---|---|---|
| Raycast | Snap to nearest (existing) | N/A | Baseline Technique |
| Aperture | Translucent wand display | Multi-step Disambiguation | Requires similar input metaphor |
| Bubble | Adaptive shape, limit and control size | N/A | Useful when less dense than Expand and SQUAD |
| Scope | Adaptive low-speed size, dynamic offset | N/A | Good for low to medium density |
| Implicit Fan | Extended Fan Angle adjustment, directional scoring, adaptive sizing | N/A | Integrate similar functionality as Expand or SQUAD |

# CHAPTER 7: DISCUSSION

Throughout the course of our research, we encountered a few topics that were worthy of discussion, although did not fit well within the relevant chapters. Here, we will discuss these topics in greater detail.

## Section 7.1: Learning Effect

The data collected as a part of our user studies was used to measure certain things that we were most interested in, such as user accuracy or speed. However, there lies a lot of potential with the data we collected to examine other metrics, such as the user's ability to learn and improve his or her performance with a specific technique or scenario.

With all of our data samples, we sought to analyze performance of a technique within a certain scenario. To compute this, we would have the user perform that scenario several times, and then compute the average. All scenarios were random for all users, so as to minimize the learning effect over all users and test cases. Despite this, there is still the opportunity to take a single user's performance with a particular technique and observe it in a single scenario for all iterations, and measure any deviation in performance as they repeated it later in the study. When all scenarios for a single user have been analyzed, the user can then be assigned a learning value that represents how much their performance improved over the course of the study, both as a whole, and for a single technique. After all users have been analyzed, then the learning value for each user per technique could be combined, revealing the extent to which each

technique offers the opportunity for learned improvement. This is a seemingly overlooked aspect of any technique that would appear to be a valuable piece of information.

## Section 7.2: Auto-Select Selection Timing

In the construction of our Auto-Select Framework (ASF), we designed it in such a way that requires the analysis of each frame to determine which technique was optimal at any moment. This is important because if the techniques are visually distinct in their pre-selection feedback and operation, then the user needs to know about it and thus this change needs to be occurring in real time. It may not always be the case that the techniques are visually similar in this manner, and the framework cannot possibly be designed to make this judgment, so we believe that it is within reason to have designed it the way we did.

If we instead imagined a configuration where all techniques involved were visually identical right up until the point of selection, then there is no longer a need to be computing the optimal technique every frame, since there is nothing to show the user that could possibly change on a frame by frame basis. This permits the optimal determination to be deferred until the user actually presses the selection trigger. The consequences of delaying the determination are mixed, and will vary by technique. Overall though, the reduction in computation could result in a performance increase, if the system was already operating at its limit or at a less than optimal frame rate. For individual techniques, any over-time sampling that is required can still be performed,

just as is done in any other case, so it is not likely that any significant impact would be seen on a single technique, but this cannot be entirely ruled out.

## Section 7.3: Participant Assessment

During the course of completing the post-questionnaires for our various user studies, we often asked the participants to rank the techniques by preference, such that a 1 indicated the highest preference and a 4 (if 4 techniques) indicated the lowest preference. The verbiage that described how to properly answer this section of the questionnaires was changed from study to study, based on observation of participants completing the section in previous questionnaires of past studies. This type of question seemed to pose a real problem for proper completion, even when the instructions were very clear about how to complete it. This casts some doubt to the validity of such a question, and warrants future investigation into how to best frame such a question.

In addition to the above mentioned issue, there is also the potential issue of bias. The order that the techniques are presented in the question could have an effect on how they are ranked. Since we utilized paper questionnaires, it was not practical to have the order randomized. If we had a digital system, then that would have been a simple and logical thing to do to improve the validity of the results. We have not done any analysis on the rankings based on the order, and likely cannot do so, since the actual desirability of the techniques would mask any subtle bias that might be present.

Another challenge with asking the users about selection technique preference when utilizing the ASF is that users are not directly aware they are using ASF. From their perspective, they are using the techniques that are contained within it. The ASF is really a logical wrapper around the techniques, meant to make the act of selection better without being directly observed. As a result, asking the user to rank their preference of ASF versus the techniques contained within is confusing at best and worthless at worst. During the post-questionnaire, we would recite a specific set of guidelines to remind the participant what each technique was, including ASF, so that they could recall well enough to validly answer the question. Despite this, we were not confident that the participants were adequately and accurately answering the question. This is not a fault in the reasoning, but rather putting them into a situation where the knowledge we are asking them to provide is not something that is easily determined or discernable from their experiences.

## Section 7.4: Continued Technique Improvement

The pursuit of novel selection techniques which improve over their predecessors is something that we believe has some value, but should be done with a certain level of awareness. The existing breadth of work in this area has already lead to a large base to work with, and should be leveraged for existing scenarios were appropriate before jumping to the conclusion that a new technique must be invented. A new selection technique should be designed only where there is a perceived need or obvious

shortcoming in the matchup between existing techniques and the scenario(s) in question.

With that being said, we do feel that there is ample reason to continue research in the field of selection techniques. As we have discussed in CHAPTER 6:, there are several reasons why a researcher might want to create a new technique. The ever increasing complexity of scenarios and new situations that users are put in will ultimately drive the desire to create selection techniques that are able to cope with the broad spectrum of conditions that the user will be placed in. The adaptability of a technique will be what sets it apart from any existing static technique, leading to an improvement in usability, if not outright performance as measured by time or accuracy. As we noted with our Scope technique in Chapter 5, users commented on the visually dynamic aspect of it, stating that they liked how it appeared to function. Based on our quantitative data, we were able to see that this did not lead to an increase in performance over similar techniques, but given the option, we believe users would take a usability improvement even if it didn't lead to a performance increase. It is our overarching opinion that new selection techniques should continue to be researched, but not with the goal of increasing raw performance. The overall usability and satisfaction with the using the technique should be the primary goal, as that is what ultimately drives the user to enjoy using it.

## Section 7.5: Selection Technique Design

Throughout our work, we have created numerous selection techniques. Some were of our own design, some were a derivative of an existing technique, and some were an

implementation of an existing technique for the sake of comparison testing. The design process for each technique varied slightly, but still had common elements that were present each time. For this section, we focus specifically on the way in which we designed new techniques and what we learned from this process.

Whenever designing a selection technique, it is important to have good motivation for doing so. In our case, we found that existing techniques seemed to be lacking in performance when it came to working across a large set of environments. Scrutinizing the techniques in use allowed us to identify features which we believed could be redesigned or improved, giving us a specific design goal that we could start with. These features were conceptualized, and then implemented. Initial testing was used to validate the correctness of our ideas, but more importantly gave valuable feedback regarding how incorrect we may have been. There are so many variables with selection, such as user ability, hardware devices, environmental conditions, and scenario complexity to name a few. The importance of iteratively testing and modifying a technique for best performance in typical conditions cannot be overstated. When performance metrics are gathered, it is important to realize that there is more to a technique than timing and accuracy. Even if performance is on par with an existing technique, it is important to remember that having choices is valuable, both in selection, and life in general.

# CHAPTER 8: CONCLUSIONS AND FUTURE WORK

In our research, we obtained a diverse set of results that relate to various aspects of selection theory and the process of building an intelligent selection framework. We have created a set of new dynamic selection techniques, a high-level selection framework, and proposed several new ways to improve upon existing dynamic selection techniques that are actionable and part of our future work. These results are not only immediate, but lead to a breadth of future topics that will further improve the understanding of selection in VEs.

## Section 8.1: Selection in Various Scenarios

Our initial research into how well various selection techniques operate in various scenarios ranging in object speed and density served as a foundation for asking tough questions about how a selection technique should operate [10]. The adaptation of a technique to its environment is not new, but the detailed scrutiny that we performed in an actual 3D VE was uncommon. The observation that some techniques were stronger than others in different scenarios was not a surprise, but it was critical in our later design of the Auto-Select framework. 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 a result, we created guidelines to assist designers with selecting between these variants and establish some preliminary guidelines for dense and dynamic 3D object selection. These guidelines are located in Section 3.4:.

## Section 8.2: Automatic Technique Assignment

The Auto-Select Framework (ASF) was created out of the desire to avoid the situation where a single technique must be implemented in a virtual environment when there is the possibility of varying conditions, some of which the technique is not really suited for. In our study, we discovered that the performance of the ASF was roughly on par with that of the dominant technique [12]. We believe that this has a lot to do with the techniques we chose to incorporate into it. The ASF itself proved that it was a valid working prototype, and could be explored more in future work. When analyzing the qualitative data, we were concerned that it would be difficult for users to understand when it was in use, and not be able to evaluate it properly. By ranking the ASF, the users are in essence ranking the techniques contained within it. For this reason, we would lean towards placing more emphasis on quantitative data whenever the ASF is utilized in a future study.

During development, it was a primary goal to make the ASF flexible and usable by other researchers. The software was written in such a manner that it only required that selection techniques that wish to be included in it implement a software interface, which

148

ensures that they perform the minimum functionality required to be interoperable. Our contribution in this regard is not only the performance aspects of the ASF, but also the manner in which it was designed. Our design has been described with enough detail such that anybody could create their own version of it and have similar results.

With the ASF, there were some issues that we noted which we thought were important to address if one was to use the framework confidently. The most important was the transition stage. When the Analyzer switches from one technique to another, it is crucial that this switch occur with the least amount of disruption to the user. To accomplish this, the techniques incorporated would need to be similar in some regards, such as visual appearance.

There is also the similar issue of user expectation. If the user is expecting the interface to act a certain way, then the techniques should not cause any unexpected behavior to occur. By this, we mean that a selection attempt should be relatively consistent across all techniques. The response that the user gets when performing a selection should be predictable, at a minimum, if it is not very similar.

This brings us to another point, which is visual feedback. In our research, we became aware that users were typically not paying attention to our feedback icon which was located near the cursor. If it is desired that the user be informed as to the potential consequences of selecting, then there needs to be a clear way to do that which they will understand. The more similar the response from the different incorporated selection techniques, the less important this feedback is. The most obvious way of performing this is by subtly modifying the selection cursor, which also benefits from designing the

cursor to resemble the expected behavior. An example of this would be a circular, encompassing cursor icon for an area cursor, and a crosshair for a point cursor. This is a trivial example, but emphasizes how the cursor can indicate how the technique works.

## Section 8.3: Dynamic Adaptation of Selection Techniques

In this work, we presented a method in which we incorporated various components of different selection techniques into a single technique, with the goal of increasing its usability across a broader diversity of scenarios. Our selection technique, Scope, showed serious promise as a viable technique, but there was more to it than just creating another technique. The picking and choosing of favorable features and incorporating them is a repeatable process that we intend on doing more of. In this specific case though, we did come to understand that the visual component of Scope was appreciated by many users, lending credibility to its design. Even if a selection technique does not necessarily perform better than another, there is still value in increasing the user satisfaction, if nothing else is gained. The manner in which Scope reacted to the user's input actions allowed them to feel more immersed in the technique, which might explain why they indicated that they enjoyed using it.

## Section 8.4: Efficacy of Existing Dynamic Selection Techniques

In our latest work, we performed a detailed analysis of existing dynamic selection techniques, focusing on their design motives, dynamic attributes, strengths, and

weaknesses (CHAPTER 6:). For each of these techniques, we discussed how it could be expanded upon, improving its performance by addressing some of its notable flaws. Where suitable, we considered how each technique could be integrated into our Auto-Select Framework for improved performance and broader deployment. We also discussed how some techniques could include a two-step selection function to improve high-density selection accuracy.

As already mentioned some in Section 6.3.2:, Aperture is a selection technique that has many features which we propose for improvement and expansion. The rare incorporation of a specific input device such as the wand opens up another dimension of ways in which Aperture can be modified. We proposed the inclusion of a translucent display, which would then permit things such as dynamic semi-physical cursors on the display. It could operate in a similar manner to the magic lens [4], but as a physical layer instead of a virtual one. This type of display would truly open the doors to innovative improvements and variations of the technique that we only scratched the surface of.

Not only did we propose the translucent display aperture, but we also included a method of performing object disambiguation in cases where the base technique fails to perform well, specifically in very dense environments. This multi-step disambiguation function could operate much like the Expand technique as we described, or in another manner such as how SQUAD operates. We believe that there are many more ways to do this in multiple steps, but we provide these two as an obvious starting point.

The Implicit Fan Cursor (IFC) was another technique that we were able to aggressively suggest feature improvements for, as we have already discussed in Section 6.3.5:. The

integration of these improvements will likely lead to a very versatile selection technique that works well in a broad spectrum of conditions similar to what we have tested in.

## Section 8.5: Future Work

Throughout the course of our work, we reached conclusions that generated new questions. We were only able to pursue some of these, which is what guided the overall path of our work. Those questions that were left unanswered are still enticing research opportunities that we would like to someday pursue.

### Section 8.5.1: Additional Dimensions of Selection

In our work, we focused primarily on two dimensions of selection: the density of objects in the scene, and the average velocity of said objects [10]. Our initial study contained a user study which consisted of several different combinations of each, in an attempt to get some clarifying data that would best highlight the true performance of the techniques being tested. Our data was solid, but still was only within the scope of those two dimensions. If we could explore other dimensions, we could gain additional insight into the performance of existing techniques. These dimensions could be determined by analyzing existing games and simulations to see what users are being expected to perform in, and finding those that are most prevalent.

One example of another dimension that could be tested is level of visibility, or brightness of an environment. Often time in games, a user is expected to select

something in a very dark environment, one where it is difficult to see in. In some cases, the user is provided a light enhancing tool, such as a flashlight, torch, or lantern. With such limited visibility, it can be expected that some techniques will fare differently than when they are in well let conditions. At first thought, any technique which contains a visual feedback mechanism to indicate the potential selection of an object will have an advantage over one that doesn't, since the user may not even see the object which they might be trying to select.

Another variant of this type of study in this example could be to define which features of a selection technique are most important in low visibility situations. Such features could include the allowing of empty selection and the presence of visual or auditory feedback. The existence of these features will likely make a significant impact on the performance in low light scenarios. We would like to test these ideas in another formal user study by implementing various techniques with such features and testing them across a variety of lighting conditions to see how strong of an impact they make, and which are most important to performance.

*Section 8.5.2: Auto-Select Framework Revisited*

In our initial design, the Auto-Select Framework (ASF) was designed to control the flow of information from the scene to the technique, while also restricting which pieces of information the techniques had at their disposal to make decisions regarding their suitability of selection. In our study, we were only concerned with and were only testing for how the selection techniques worked across various levels of density and velocity,

so it was not a problem at the time. For all future applications where other developers would use the ASF with a variety of techniques, if would be much more appropriate if the framework permitted each technique to directly access the scene when making the determination of suitability. This is a good idea for several reasons. First, allowing direct access would simplify the ASF, making it less restrictive and thus more useful and practical. Secondly, it would permit each technique to use whatever feature of the scene it needed to, which would allow for a more expressive utilization of the ASF by including a broader variation of techniques. There may be more reasons, but these two alone are sufficient enough to warrant a redesign, in our opinion.

In addition to the redesign of the internal operation of the ASF, we would also like to explore the possibility of creating a well-defined set of guidelines for designing techniques that are to operate with the ASF. The most important aspects of the ASF is the internal algorithm within each technique that determines how suitable it is, given the existing conditions in the scene. The value that is computed must be standardized, or else the values returned by different techniques will not be comparable to each other. The guidelines that we would establish would not only define the software specifics for how to integrate the selection technique into the ASF, but also contain a standard set of scenarios for which to test the selection technique against, so that the suitability index could be normalized across all techniques. A naïve unit of measurement for this index could simply be based on the expected success rate of selection, where 0.0 is always incorrect, 0.5 is correct selection, and 1.0 is always correct. In this case though, it should be noted that typically, the software does not have a single target that is any

more correct than any other. Despite this, in the normalization process of each technique, the values obtained would still be based on the user selecting a correct object, and thus the accuracy of the technique would be based still on a measurement of accuracy of the user's actions with selecting an intended object.

Once we have completed the redesign of the ASF, we would like to incorporate many different techniques into it that were not tested initially. With our results from Chapter 6, we could take the selection techniques reviewed and make the proposed modifications, then test them in the ASF to evaluate any performance benefit, as well as validate the changes made to the ASF itself. We could also experiment with the effect that the quantity of potential selections techniques has on performance. We originally tested just two, but it would be interesting to test three or more, and include additional dimensions of variability to the environments, such as our proposed idea of testing the level of visibility.

*Section 8.5.3: Selection Technique Improvements*

In Chapter 6, we studied several dynamic selection techniques. For Aperture, Scope, and Implicit Fan Cursor, we came up with some ideas for improving the techniques. In the case of Aperture, the modifications would require the construction of a translucent secondary display, which would be quite an investment in both time and cost. While it would be a good research project, it would require the most investment to fully explore. For Scope and Fan, the proposed changes to design would be all software based, which is relatively easy and would not require considerable development time.

Therefore, we would like to take those two and make our suggested modifications and test them against the baseline version of each. The test would be done across a variety of scenarios, similar to our past studies.

With the Implicit Fan Cursor as a basis, we would like to take it and modify several properties and methods. First, we would like to modify the way in which it generates the fan shape. We believe that the total open angle of the fan should begin at 360 when below a certain speed threshold, and then slowly narrow to some minimum angle when a certain speed is reached. Second, we would like to incorporate a scoring function that operates over time to better identify which target the user is expected to select. Third, we would like to dynamically alter the throw, or length of the fan in relation to cursor velocity. At higher speeds, the length would grow, encompassing a greater area ahead of the cursor. Finally, we would like to configure the scoring algorithm that we implement with it to place a greater emphasis on objects that are more in a direct path, and place a lesser emphasis on those objects that are at more of an angle from the anticipated path of movement. With these changes, we believe that the performance of the technique could be vastly improved.

APPENDIX A:
VR2012 PRE-QUESTIONNAIRE

Date: _____/_____/_____     Participant Number: _____

**Selection User Study**
**ISUE Lab, Computer Science Division**
**University of Central Florida**
**Pre-Questionnaire**

## Demographic Questions

Gender (please circle one):  Male / Female     Age: _____     Major: _____

Do you require any eye-aids (glasses, contacts) to play video games? Yes / No
If yes, will you be wearing them while performing the study?     Yes / No     *(Please do, if you need them)*

## Gaming Questions

*Please answer the following questions by circling the most appropriate number.*

1. About how often do you play video games?
   Never   Rarely   Once a month   Once a week   Couple days a week   Practically every day

2. About what percent of your time playing games is with a game that is 3D in nature?
   *examples include: Call of Duty, Madden, Gran Turismo, God of War, Little Bit Planet, Super Mario Galaxy, Need for Speed*
   0%     25%     50%     75%     100%

3. About what percent of your time playing games is spent playing 3D shooter games?
   *examples include: Call of Duty, Battlefield series, Medal of Honor, Half-Life, Crysis, Just Cause*
   0%     25%     50%     75%     100%

4. Do you have any experience using the Sony Move controller? (seen at right)
   Yes     No     Not Sure

5. Where do you rank your general gaming skill?
   Beginner     Amateur     Average     Very good     Expert

6. How comfortable would you say you are with using a controller to point to desired objects on the screen?
   *examples include: Using Nintendo Wii controller to navigate home screen, or choosing something on PS3 with PS Move controller*
   Not at all     Somewhat     Modestly     Very     Extremely

7. Do you have any flexibility and/or pain issues with your dominant arm and/or hand?
   No     Yes, a little     Yes, a lot

158

APPENDIX B:
VR2012 POST-QUESTIONNAIRE

**Selection User Study**
**ISUE Lab, Computer Science Division**
**University of Central Florida**
**Post-Questionnaire**

## Gaming Questions

*Please answer the following questions by circling the most appropriate **number**.*

1. How would you rate the Raycast technique, with respect to the following characteristics?

   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

2. How would you rate the Zoom technique, with respect to the following characteristics?

   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

3. How would you rate the Expand technique, with respect to the following characteristics?

   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

4. How would you rate the SQUID technique, with respect to the following characteristics?

   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

5. How adequate do you feel the 3D interface was at allowing you to make proper selections?

   very inadequate    1    2    3    4    5    6    7    very adequate

6. How adequate do you feel the time allotted for practice was?

   very inadequate    1    2    3    4    5    6    7    very adequate

7. How comfortable were you with using the Sony Move Controller for this exercise?

   very uncomfortable    1    2    3    4    5    6    7    very comfortable

8. Rank the four selection techniques in order of desired use, with 1 being most desired and 4 being least desired.
   *Note: Each number can only be chosen once*

   | Raycast | 1 | 2 | 3 | 4 | (Simple point and select) |
   | Zoom | 1 | 2 | 3 | 4 | (Point, camera zooms in, then you select) |
   | Expand | 1 | 2 | 3 | 4 | (Objects slide into view, you select, then they slide back) |
   | SQUID | 1 | 2 | 3 | 4 | (Objects appear in X grid and you select (multiple) times) |

9. When determining how much you like using a selection technique, how much influence does ease-of-use have on your decision?

   none at all    1    2    3    4    5    6    7    a lot

10. When determining how much you like using a selection technique, how much influence does speed have on your decision? *By this, I mean the amount of time required to make an accurate selection.*

   none at all    1    2    3    4    5    6    7    a lot

160

APPENDIX C:
VR 2013 PRE-QUESTIONNAIRE

**Selection User Study**
**ISUE Lab, Computer Science Division**
**University of Central Florida**
**Pre-Questionnaire**

## Demographic Questions

Gender (please circle one):  Male / Female        Age: _____        Major: _____

Do you require any eye-aids (glasses, contacts) to play video games? Yes / No
If yes, will you be wearing them while performing the study?        Yes / No   *(Please do, if you need them)*

## Gaming Questions

*Please answer the following questions by circling the most appropriate* **choice**.

1. About how often do you play video games?
   Never   Rarely   Once a month   Once a week   Couple days a week        Practically every day

2. About what percent of your time playing games is with a game that is 3D in nature?
   *examples include: Call of Duty, Madden, Gran Turismo, God of War, Little Bit Planet, Super Mario Galaxy, Need for Speed*
   0%        25%        50%        75%        100%

3. About what percent of your time playing games is spent playing 3D shooter games?
   *examples include: Call of Duty, Battlefield series, Medal of Honor, Half-Life, Crysis, Just Cause, Halo*
   0%        25%        50%        75%        100%

4. Do you have any experience using the Sony Move controller? (seen at right)
   Yes        No        Not Sure

5. Where do you rank your general gaming skill?
   Beginner        Amateur        Average        Very good        Expert

6. How comfortable would you say you are with using a controller to point to desired objects on the screen?
   *examples include: Using Nintendo Wii controller to navigate home screen, or choosing something on PS3 with PS Move controller*
   Not at all        Somewhat        Modestly        Very        Extremely

7. Do you have any flexibility and/or pain issues with your dominant arm and/or hand?
   No        Yes, a little        Yes, a lot

162

APPENDIX D:
VR 2013 POST-QUESTIONNAIRE

**Selection User Study, Post-Questionnaire**
**ISUE Lab, Computer Science Division, University of Central Florida**

## Gaming Questions

*Please answer the following questions by circling the most appropriate **number**.*

1. How adequate do you feel the 3D interface was at allowing you to make proper selections?

   very inadequate     1    2    3    4    5    6    7     very adequate

2. How adequate do you feel the time allotted for practice was?

   very inadequate     1    2    3    4    5    6    7     very adequate

3. How comfortable were you with using the Sony Move Controller for this exercise?

   very uncomfortable     1    2    3    4    5    6    7     very comfortable

4. How would you rate the Raycast technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

5. How would you rate the Expand technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

6. How would you rate the Auto-Select technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

7. Rank the four selection techniques in order of desired use, with 1 being most desired and 4 being least desired.

   *Note: Each number can only be chosen **once***

   | | | | | |
   |---|---|---|---|---|
   | Raycast | 1 | 2 | 3 | (Simple point and select) |
   | Expand | 1 | 2 | 3 | (Objects slide into view, you select, then they slide back) |
   | Auto-Select | 1 | 2 | 3 | (Technique is chosen for you) |

8. How <u>helpful</u> was the feedback which indicated which selection technique would be used with Auto-Select?

   none at all     1    2    3    4    5    6    7     extremely

9. When determining how much you like using a selection technique, how much influence does <u>ease-of-use</u> have on your decision?

   none at all     1    2    3    4    5    6    7     a lot

10. When determining how much you like using a selection technique, how much influence does <u>speed</u> have on your decision? *By this, I mean the amount of time required to make an accurate selection.*

    none at all     1    2    3    4    5    6    7     a lot

APPENDIX E:
3DUI 2013 PRE-QUESTIONNAIRE

**Selection User Study**
**ISUE Lab, Computer Science Division**
**University of Central Florida**
**Pre-Questionnaire**

## Demographic Questions

Gender (please circle one): Male / Female      Age: _____      Major: _____

Do you require any eye-aids(glasses, contacts) to play video games? Yes / No
If yes, will you be wearing them while performing the study?      Yes / No   *(Please do, if you need them)*

## Gaming Questions

*Please answer the following questions by circling the most appropriate* **choice**.

1. About how often do you play video games?
   Never   Rarely   Once a month   Once a week   Couple days a week   Practically every day

2. About what percent of your time playing games is with a game that is 3D in nature?
   *examples include: Call of Duty, Madden, Gran Turismo, God of War, Little Bit Planet, Super Mario Galaxy, Need for Speed*
   0%   25%   50%   75%   100%

3. About what percent of your time playing games is spent playing 3D shooter games?
   *examples include: Call of Duty, Battlefield series, Medal of Honor, Half-Life, Crysis, Just Cause, Halo*
   0%   25%   50%   75%   100%

4. Do you have any experience using the Sony Move controller? (seen at right)
   Yes   No   Not Sure

5. Where do you rank your general gaming skill?
   Beginner   Amateur   Average   Very good   Expert

6. How comfortable would you say you are with using a controller to point to desired objects on the screen?
   *examples include: Using Nintendo Wii controller to navigate home screen, or choosing something on PS3 with PS Move controller*
   Not at all   Somewhat   Modestly   Very   Extremely

7. Do you have any flexibility and/or pain issues with your dominant arm and/or hand?
   No   Yes, a little   Yes, a lot

# APPENDIX F:
# 3DUI 2013 POST-QUESTIONNAIRE

*Please answer the following questions by circling the most appropriate NUMBER.*

1. How adequate do you feel the 3D interface was at allowing you to make proper selections?

   very inadequate    1   2   3   4   5   6   7    very adequate

2. How adequate do you feel the time allotted for practice was?

   very inadequate    1   2   3   4   5   6   7    very adequate

3. How comfortable were you with using the Sony Move Controller for this exercise?

   very uncomfortable    1   2   3   4   5   6   7    very comfortable

4. How would you rate the Bendcast technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

5. How would you rate the Expand technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

6. How would you rate the Auto-Select technique, with respect to the following characteristics?

   | | | | | | | | | |
   |---|---|---|---|---|---|---|---|---|
   | usability: | very difficult | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very easy |
   | speed: | very slow | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very fast |
   | accuracy: | very inaccurate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | very accurate |

7. Rank the four selection techniques in order of desired use, with 1 being most desired and 3 being least desired.
   *Note: Each number can only be chosen once*

   | | | | | |
   |---|---|---|---|---|
   | Bendcast | 1 | 2 | 3 | (Simple point and select) |
   | Expand | 1 | 2 | 3 | (Objects slide into view, you select, then they slide back) |
   | Auto-Select | 1 | 2 | 3 | (Technique is chosen for you) |

8. How helpful was the feedback which indicated which selection technique would be used with Auto-Select?

   none at all    1   2   3   4   5   6   7    extremely

9. When determining how much you like using a selection technique, how much influence does ease-of-use have on your decision?

   none at all    1   2   3   4   5   6   7    a lot

10. When determining how much you like using a selection technique, how much influence does speed have on your decision? *By this, I mean the amount of time required to make an accurate selection.*

    none at all    1   2   3   4   5   6   7    a lot

APPENDIX G:
HUMAN SUBJECT CONSENT FORM (3 PAGES)

# University of Central Florida

## 3D Selection Techniques: Studying the Speed and Accuracy of Various 3D Selection Techniques

### Informed Consent for an Adult in a Non-Exempt Research Study

Principal Investigator: Jeff Cashion
Co-Investigator: Dr. Joseph LaViola

### Introduction

Researchers at the University of Central Florida (UCF) study many topics. To do this we need the help of people who agree to take part in a research study. You are being invited to take part in a research study which will include about 24 people at UCF. You have been asked to take part in this research study because you are an able person who responded to a request for participants. You must be 18 years of age or older to be included in the research study.

The person doing this research is Jeff Cashion of the UCF Computer Science Department. Because the researcher is a PhD student, he is being guided by Dr. Joseph LaViola, a UCF faculty supervisor in Computer Science.

### What should you know about a research study?

- Someone will explain this research study to you.
- A research study is something that you volunteer for.
- You are not required to participate.
- You should only participate in this study because you want to.
- You can choose not to take part in the research study.
- You can agree to participate now and change your mind later.
- Your decision to participate or not will not be held against you.
- You are free to ask any and all questions that you like before you make your decision.

### Purpose of the research study

The goal of this user study is to better evaluate the performance characteristics of several selection techniques. You will use a Sony Move Controller to control an on-screen cursor which will be used to select objects on the screen. Different selection techniques will be tested, and then compared by the researchers to determine which was best suited for the job. This research will give insight into how to better perform selection techniques so that fewer errors will be made while allowing accurate selection to require less time.

# University of Central Florida

**What will you be asked to do in the study?**

You will first be given a pre-questionnaire for collecting demographic information and video game experience. Your name will not be recorded; only a number for maintaining order in the results. You will then be taken to the area where you will be performing the selection tests, with only yourself and the researcher present.

In the study, you will be testing several selection techniques. The order that the techniques are presented will be randomized. When testing each technique, you will be selecting objects in nine scenarios, which are also randomized. When first starting, you will be given one minute to practice using the system. Basic instructions will be given to inform you how to use the controls. When first placed at a new scene, you will have five seconds to observe the scene before being allowed to begin your selection. This is to get you acquainted with the scene and to reduce enhancing effects from memorization of a scene.

After all tests have been evaluated, you will be asked to fill out a post-questionnaire which will aid the researchers in determining what technique you preferred and what your level of enjoyment was.

**Location:** Interactive Systems & User Experience Lab - Harris Engineering Center, Room 208

**Time required:** We expect that you will be in this research study for a maximum of 30 minutes.

**Risks**

You might experience slight fatigue after using the Sony Move Controller for an extended period of time, which is approximately 10 to 20 minutes.

**Benefits**

There are no direct benefits to you other than having a good time. An indirect benefit will hopefully be an improvement in future selection technique implementations.

**Compensation or payment**

You will be paid $10 for your time, to be paid at the completion of the session.

**Decision to Participate, Opt Out and Right to Quit at Any Time**

A decision to quit the study will not affect any future relationship with the Principal Investigator, the experimenters, the Interactive Systems and User Experience Lab, the Computer Science Department, the School of Electrical Engineering and Computer Science, or University of Central Florida. You are not required to answer any questions you do not wish to.

# University of Central Florida

At any time during the experiment, you have a right to withdraw consent without any consequence. You must complete the entire experiment to receive $10.

**Study contact for questions about the study or to report a problem**

If you have any questions, concerns, complaints, or think the research has hurt you, talk to Jeff Cashion at jcashion@knights.ucf.edu or Faculty Advisor, Dr. Joseph LaViola at (407) 882-2285 or jjl@eecs.ucf.edu

**IRB contact about your rights in the study or to report a complaint**

Research at the University of Central Florida involving human participants is carried out under the oversight of the Institutional Review Board (UCF IRB). This research has been reviewed and approved by the IRB. For information about the rights of people who take part in research, please contact: Institutional Review Board, University of Central Florida, Office of Research & Commercialization, 12201 Research Parkway, Suite 501, Orlando, FL 32826-3246 or by telephone at (407) 823-2901. You may also talk to them for any of the following:

- Your questions, concerns, or complaints are not being answered by the research team.
- You cannot reach the research team.
- You want to talk to someone besides the research team.
- You want to get information or provide input about this research.

# APPENDIX H:
# HUMAN RESEARCH PERMISSION FORM

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

## Approval of Human Research

From: **UCF Institutional Review Board #1**
**FWA00000351, IRB00001138**

To: **Jeffrey A. Cashion**
Date: **July 17, 2014**

Dear Researcher:

On 07/17/2014, the IRB approved the following human participant research until 7/16/2015 inclusive:

| | |
|---|---|
| Type of Review: | IRB Continuing Review Application Form |
| | Expedited Review |
| Project Title: | 3D Selection Techniques: Studying the speed and accuracy of various 3D selection techniques. |
| Investigator: | Jeffrey A Cashion |
| IRB Number: | SBE-11-07814 |
| Funding Agency: | |
| Grant Title: | |
| Research ID: | N/A |

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 7/16/2015, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

*Kanielle Chap*

# LIST OF REFERENCES

[1]  F. Argelaguet and C. Andujar, "Efficient 3D Pointing Selection in Cluttered Virtual Environments," *IEEE Computer Graphics and Applications,* vol. 29, no. 6, pp. 34-43, 2009.

[2]  F. Argelaguet and C. Andujar, "A survey of 3D object selection techniques for virtual environments," *Computers & Graphics,* vol. 37, no. 3, pp. 121-136, 2013.

[3]  R. Balakrishnan, "'Beating'' Fitts' law: virtual enhancements," *International Journal on Human-Computer Studies,* vol. 61, pp. 857-874, 2004.

[4]  E. A. Bier, M. C. Stone, K. Pier, W. Buxton and T. D. DeRose, "Toolglass and Magic Lenses: The See-through Interface," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, Anaheim, C, 1993.

[5]  R. Blanch, Y. Guiard and M. Beaudouin-Lafon, "Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation," *CHI,* vol. 6, no. 1, pp. 519-526, 2004.

[6]  D. Bowman and L. Hodges, "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments," *Symposium on Interactive 3D Graphics,* pp. 35-38, 1997.

175

[7]  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.,* vol. 11, no. 4, pp. 404-424, 2002.

[8]  D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev, 3D User Interfaces: Theory and Practice, Addison-Wesley, 2004.

[9]  D. Bowman, Interaction Techniques for Common Tasks In Immersive Virtual Environments, (Doctoral dissertation), 1999.

[10] J. Cashion, C. Wingrave and J. J. LaViola, "Dense and Dynamic 3D Selection for Game-Based Virtual Environments," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 18, no. 4, pp. 634-642, 2012.

[11] J. Cashion and J. J. LaViola, "Dynamic adaptation of 3D selection techniques for suitability across diverse scenarios," in *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, Minneapolis, MN, 2014.

[12] J. Cashion, C. Wingrave and J. J. LaViola, "Optimal 3D Selection Technique Assignment Using Real-Time Contextual Analysis," *3D User Interfaces (3DUI), 2013 IEEE Symposium on,* 2013.

[13] A. Celentano and F. Pittarello, "Observing and Adapting User Behavior in Navigational 3D Interfaces," *Proceedings of the working conference on Advanced visual interfaces (AVI),* pp. 275-282, 2004.

176

[14] O. Chapuis, J. Labrune and E. Pietriga, "DynaSpot: Speed-Dependent Area Cursor," *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09),* pp. 1391-1400, 2009.

[15] O. Chapuis and P. Dragicevic, "Effects of Motor Scale, Visual Scale, and Quantization on Small Target Acquisition Difficulty," *ACM Transactions on Computer-Human Interaction,* vol. 18, no. 3, 2011.

[16] A. Cockburn and P. Brock, *Human on-line response to visual and motor target expansion,* 2006, pp. 81-97.

[17] A. Cockburn and A. Firth, "Improving the Acquisition of Small Targets," in *British HCI Conference*, pp 181-196, 2003.

[18] E. R. Crossman and P. J. Goodeve, "Feedback control of hand-movement and Fitts' law," *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology,* vol. 35, no. 2, pp. 251-278, 1983.

[19] N.-T. Dang, "A Survey and Classification of 3D Pointing Techniques," *IEEE International Conference on Research, Innovation and Vision for the Future,* pp. 71-80, 2007.

[20] G. De Haan, M. Koutek and F. Post, "IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection," *Eurographics Workshop on Virtual Environments,* pp. 201-209, 2005.

[21] M. A. Finch, J. G. Phillips and J. W. Meehan, "Cursor type and response conflict in graphical user interfaces," *Interacting with Computers,* vol. 19, no. 1, pp. 113-120, 2007.

[22] P. Fitts, "The information capacity of the human motor system is controlled by the amplitude of movement," *J. Exp. Psychol.,* vol. 6, no. 47, pp. 381-391, 1954.

[23] A. S. Forsberg, J. J. LaViola Jr., L. Markosian and R. C. Zeleznik, "Seamless Interaction in Virtual Reality," *IEEE Comput. Graph. Appl.,* vol. 17, no. 6, pp. 6-9, 1997.

[24] A. Forsberg, K. Herndon and R. Zeleznik, "Aperture Based Selection for Immersive Virtual Environments," *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96),* pp. 95-96, 1996.

[25] S. Frees, "Context-Driven Interaction in Immersive Virtual Environments," *Virtual Reality,* vol. 14, no. 4, pp. 277-290, 2010.

[26] T. Grossman and R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 281-290, 2005.

[27] Y. Guiard, "The Problem of Consistency in the Design of Fitts' Law Experiments: Consider either Target Distance and Width or Movement Form and Scale," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*

*(CHI '09),* pp. 1809-1818, 2009.

[28] C. Gutwin, "Improving Focus Targeting in Interactive Fisheye Views," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Minneapolis, Minnesota, USA, 2002.

[29] M. Hertzum and K. Hornbæk, "Input techniques that dynamically change their cursor activation area: A comparison of bubble and cell cursors," *International Journal of Human-Computer Studies,* vol. 65, no. 10, pp. 833-851, 2007.

[30] K. Hinckley, R. Pausch, J. C. Goble and N. F. Kassell, "A Survey of Design Issues in Spatial Input," *Proceedings of the 7th annual ACM symposium on User interface software and technology,* pp. 213-222, 1994.

[31] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics,* vol. 6, no. 2, pp. 65-70, 1979.

[32] T. Igarashi and K. Hinckley, "Speed-dependent automatic zooming for browsing large documents," *Proceedings of ACM UIST,* pp. 139-148, 2000.

[33] P. Kabbash and B. William, "The "prince" technique: Fitts' law and selection using area cursors," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, Denver, Colorado, USA, 1995.

[34] M. Kobayashi and T. Igarashi, "Ninja cursors: using multiple cursors to assist target acquisition on large screens," *Proc. CHI '07,* pp. 949-958, 2008.

[35] R. Kopper, F. Bacim and D. Bowman, "Rapid and accurate 3D selection by progressive refinement," *IEEE Symposium on 3D User Interfaces (3DUI),* pp. 67-74, 2011.

[36] E. Lank, Y.-C. N. Cheng and J. Ruiz, "Endpoint prediction using motion kinematics," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 637-646, 2007.

[37] J. Laukkanen, P. Isokoski and K.-J. Räihä, "The cone and the lazy bubble: two efficient alternatives between the point cursor and the bubble cursor," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 309-312, 2008.

[38] J. Liang and M. Green, "JDCAD: a highly interactive 3D modeling system," *Proceedings of Third International Conference on CAD and Computer Graphics,* pp. 217-222, 1994.

[39] I. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction," *Human Computer Interaction,* vol. 7, pp. 91-139, 1992.

[40] M. McGuffin and R. Balakrishnan, "Acquisition of Expanding Targets," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02),* pp. 57-64, 2002.

[41] M. McGuffin and R. Balakrishnan, "Fitts' Law and Expanding Targets: Experimental

Studies and Designs for User Interfaces," *ACM Transactions on Computer-Human Interaction,* vol. 12, no. 4, pp. 388-422, 2005.

[42] A. Mossel, B. Venditti and H. Kaufmann, "DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments," in *Laval Virtual VRIC*, Laval, France, 2013.

[43] J. Octavia, C. Raymaekers and K. Coninx, "Adaptation in Virtual Environments: Conceptual Framework and User Models," *Multimedia Tools and Applications,* pp. 1-22, 2010.

[44] 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.

[45] M. Ortega, "Hook: Heuristics for Selecting 3D Moving Objects in Dense Target Environments," *3D User Interfaces (3DUI), 2013 IEEE Symposium on,* 2013.

[46] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik and M. R. Mine, "Image Plane Interaction Techniques in 3D Immersive Environments," in *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island, USA, 1997.

[47] I. Poupyrev, M. Billinghurst, S. Weghorst and T. Ichikawa, "The go-go interaction technique: non-linear mapping for direct manipulation in VR," *Proceedings of the*

*9th annual ACM symposium on User interface software and technology,* pp. 79-80, 1996.

[48] I. Poupyrev, S. Webhorst, M. Billinghurst and T. Ichikawa, "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques," *EUROGRAPHICS,* vol. 17, no. 3, 1998.

[49] P. Quinn, A. Cockburn and K.-J. Raiha, "On the Costs of Multiple Trajectory Pointing Methods," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, 2011.

[50] K.-J. Raiha and O. Spakov, "Disambiguating Ninja Cursors with Eye Gaze," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, MA, USA, 2009.

[51] D. A. W. Rosa and H. H. Nagel, "Selection Techniques for Dense and Occluded Virtual 3D Environments, supported by Depth Feedback," *International Conferance of the Chilean Computer Science Society,* pp. 218-225, 2010.

[52] "Sony Move.Me SDK," 2012. [Online]. Available: https://us.playstation.com/ps3/playstation-move/move-me/.

[53] W. R. Soukoreff and S. I. MacKenzie, "Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI," *Int. J. Hum.-Comput. Stud.,* vol. 61, no. 6, pp. 751-789, 2004.

[54] A. Steed, "Towards a General Model for Selection in Virtual Environments," *IEEE Symposium on 3D User Interfaces (3DUI),* pp. 103-110, 2006.

[55] F. Steinicke, T. Ropinski and K. Hinrichs, "Object selection in virtual environments with an improved virtual pointer metaphor," *International Conference on Computer Vision and Graphics (ICCVG),* pp. 320-326, 2004.

[56] X. Su, O. Au and R. Lau, "The Implicit Fan Cursor: A Velocity Dependent Area Cursor," in *CHI 2014*, Toronto, Canada, 2014.

[57] "Unity 3D Game Development Tool," 2012. [Online]. Available: http://unity3d.com.

[58] 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),* pp. 115-122, 2007.

[59] L. Vanacken, T. Grossman and K. Coninx, "Multimodal selection techniques for dense and occluded 3D virtual environments," *International Journal of Human-Computer Studies,* vol. 67, no. 3, pp. 237-255, 2009.

[60] G. Wang and M. McGuffin, "Pop-up Depth Views for Improving 3D Target Acquisition," in *Proceedings of Graphics Interface 2011*, 2011.

[61] C. Wingrave, "3DUI Flavors Beyond Vanilla," *SEARIS Workshop,* 2009.

[62] C. Wingrave and D. Bowman, "Baseline Factors for Raycasting Selection,"

*Proceedings of Virtual Reality International,* 2005.

[63] C. Wingrave, D. Bowman and N. Ramakrishnan, "Towards Preferences in Virtual Environment Interfaces," *Proceedings of the Eurographics Workshop on Virtual Environments,* pp. 63-72, 2002.

[64] C. Wingrave, R. Tintner, B. Walker, D. Bowman and L. Hodges, "Exploring Individual Differences in Raybased Selection; Strategies and Traits," *Proceedings of the 2005 IEEE Conference on Virtual Reality (VR '05),* pp. 163-170, 2005.

[65] J. Wonner, J. Grosjean, A. Capobianco and D. Bechmann, "Starfish: a selection technique for dense virtual environments," *Proceedings of the 18th ACM symposium on Virtual reality software and technology,* pp. 101-104, 2012.