Optimal 3D Selection Technique Assignment Using Real-Time Contextual Analysis

Jeffrey Cashion[*]
University of Central Florida

Chadwick Wingrave[†]
University of Central Florida

Joseph J. LaViola Jr.[‡]
University of Central Florida

ABSTRACT

Selection in 3D virtual environments can vary wildly depending on the context of the selection. Various scene attributes such as object velocity and scene density will likely impact the user's ability to accurately select an object. While there are many existing 3D selection techniques that have been well studied, they all tend to be tailored to work best in a particular set of conditions, and may not perform well when these conditions are not met. As a result, designers must compromise by taking a holistic approach to choosing a primary technique; one which works well overall, but is possibly lacking in at least one scenario.

We present a software framework that allows a flexible method of leveraging several selection techniques, each performing well under certain conditions. From these, the best one is utilized at any given moment to provide the user with an optimal selection experience across more scenarios and conditions. We performed a user study comparing our framework to two common 3D selection techniques, Bendcast and Expand. We evaluated the techniques across three levels of scene density and three levels of object velocity, collecting accuracy and timing data across a large sample of participants. From our results, we were able to conclude that our auto-selection technique approach is promising but there are several characteristics of the auto-selection process that can introduce drawbacks which need to be addressed and minimized.

Keywords: Interaction techniques, 3D object selection, dense and dynamic environments.

Index Terms: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques;

1 INTRODUCTION

The level of detail portrayed in both 3D games and virtual environments (VE) is constantly increasing. A user's ability to adequately perceive their surroundings is becoming ever more important, and is something that developers should be focusing on with great care. One of the primary tasks in any such environment is 3D object selection [3]. With regard to 3D selection, much work has been done in developing new and efficient techniques to tackle many different types of scenarios [3]. It is generally the case that any single technique is designed to perform well only under certain conditions.

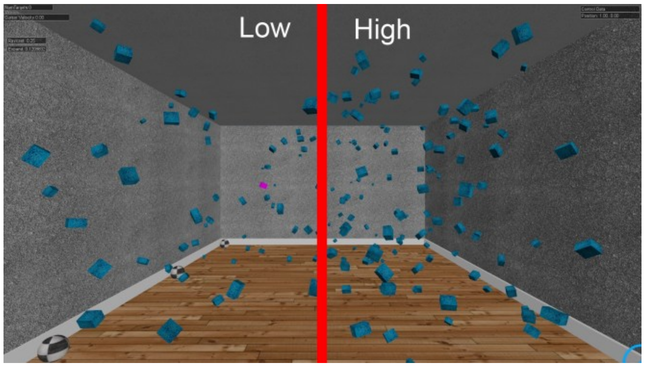

Figure 1: Low (Left) & High (Right) Density Scenario

Our work focuses on addressing this problem of compromise by exploring a method which would allow the utilization of several selection techniques, each operating where it is best suited. We are especially interested in dense and dynamic environments [4], but our work could apply to other environments as well. Our two metrics of concern are accuracy and speed, which is how we validate the correctness of our optimal selection technique assignment algorithms. Our approach involves the design of a framework which allows many selection techniques to be considered by a high-level analyzer which polls all registered techniques, asking each how well it would perform, given the instantaneous environmental conditions. From this analysis, the best available technique is chosen. As a starting point, we began by using two techniques which perform quite differently under similar conditions, and testing them across a variety of scenarios to analyze their performance. We use the term, "Auto-Select" to refer to the use of our framework for determining the optimal selection technique. We performed an evaluation of Auto-Select, comparing it to Bendcast and Expand individually. 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.

2 RELATED WORK

3D selection has been studied in great detail, and a large variety of selection techniques have been designed [4] [6]. These techniques vary in their implementation, but are all generally concerned with decreasing selection time and improving accuracy. Fitts' Law is the underlying principle of how difficult selection is when the target is static, and is used extensively to judge various selection techniques [11]. Some have even gone so far as to attempt to computationally define what a selection technique should be [13].

Fitts' Law was designed with 2D interfaces in mind, and isn't the only factor when determining how successful a user will be at selection. The experience level of each user, along with certain personal traits, play a significant role in how they will perform with selection tasks [15]. Feedback is another critical component to selection which serves to aid the user in understanding their environment [2]. It can act as a guide, steering the user towards

[*] email: jcashion@knights.ucf.edu
[†] email: cwingrav@eecs.ucf.edu
[‡] email: jjl@eecs.ucf.edu

IEEE Symposium on 3D User Interfaces 2013
16-17 March, Orlando, FL, USA
978-1-4673-6098-2/13/$31.00 ©2013 IEEE

the correct action by both positive and negative reinforcement. The caveat of this is the potential for unnecessary delays as the user attempts to comprehend the feedback itself. In cases of simple selection, this has been shown to be the case [15].

Any selection technique that does not attempt to adapt to its environment is said to be static. These have been surpassed by more dynamic techniques that seek to bend the rules of Fitts' Law by taking advantage of environmental information [1] [7]. One such example is Dynaspot [5], which varies the size of the cursor as the velocity increases, which is designed to counteract the increase of difficulty that accompanies a moving cursor. Other selection techniques, such as SQUAD [9] and Expand [4], use a multi-step iterative process to aid in target selection. Such efforts seek to improve the flexibility of a single technique, but cannot address the larger problem, which requires addressing the needs of dramatically different selection conditions.

Recent research has been performed in context-based decision making [8]. Frees defines several context components which selection techniques use as a basis for determining their suitability of use. He also proposes a framework for permitting developers to cleanly include their techniques with little burden. Adaptive frameworks have also been proposed which incorporate user preferences and behaviour into a conceptual framework for aiding in guiding interaction techniques [12]. Further modifications to the techniques themselves are suggested, such as modifying parameters and/or feedback mechanisms used. Our goal is to incorporate several features of both works, while adding our own evaluation data to test the real-world effectiveness of such an idea.

3 Baseline 3D Selection Techniques

We chose to test Bendcast and Expand since they provide a wide spectrum of performance in different scenarios. We chose to use only two techniques so that we could keep the complexity of our study to a manageable level. Only objects lying within the circular bounds of the cursor are eligible for selection.

3.1 Bendcast

Bendcast was based off of the traditional Raycast technique, with the one difference that it selects the object closest to the center of the cursor, giving the user greater selection flexibility and reduced stress. This technique works best when fewer objects are located within the cursor. It featured a simple point and click experience that should be easily understandable to all users. Our implementation calculated the distance from the center of each object to the center of the cursor, which was acceptable under our test conditions.

3.2 Expand

Expand is a two-step technique which performs well in a large variety of scenarios [4]. It was designed using an iterative process which is based on the idea of flavors [14]. The first step involves taking all objects that are inside of the cursor when the selection is triggered and placing them into a virtual grid which is tangent to the user. Objects are gathered using a Conecast [10], done by random sampling with raycasting. The objects brought into the virtual grid are actually clones of the originals, so that the original objects are left undisturbed. The virtual grid spaces out the objects in such a manner that is beneficial to selection, minimizing any occlusion and increasing selectable area. Once the expansion is complete, the user may then select from this grid.

4 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 multiple selection techniques to determine the optimal one, given any conditions. Its accuracy hinges on the accuracy of the independent algorithms within each technique. This framework was designed with the goal of allowing designers to create their own selection techniques and plugging them into the Analyzer without any dependency issues.

4.1 Framework

The core component of the framework is the Analyzer. All selection techniques must first register with the Analyzer to be utilized. The Analyzer polls all registered selection techniques, asking how suitable they are, given the provided conditions. The optimal technique is then chosen and made active. Whenever selection is attempted, the Analyzer triggers the active selection technique to perform a selection. While the active technique is selecting, the Analyzer is effectively suspended. When the selection is complete, it reports back to the Analyzer, which in turn passes the results back to the software.

Each selection technique is required to implement a specific interface which permits the Analyzer to inquire about its suitability and to command it to perform a selection. Determining the suitability index for each technique is done by independent algorithms for each technique. 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. Also included in the framework is a method responsible for reporting which image should be used for our feedback indicator, explained in Sect. 4.4.

Selection techniques utilized are expected to have little or no visual feedback when in the pre-selection state (active, not selecting). If this were not the case, then a transition from one to another would cause this pre-selection information to suddenly change, possibly causing confusion. We purposely chose two techniques that follow this guideline so as to avoid any problems. If techniques were used that featured significant pre-selection feedback, it would be up to the developer to ensure that transitions appear smooth and intuitive to the user.

4.2 Suitability Index Criterion

When establishing an algorithm for determining the optimal selection technique, one has to extract information from the scene. For our research, we used two metrics: the cursor velocity and number of objects inside of it. These 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. We passed in this information as parameters to each selection technique, but we could have also simply accessed the entire scene directly from the techniques for full access to all available information.

4.3 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. The suitability is computed on a per-frame basis, and does not have any memory. The idea of using a floating average or some advanced learning algorithm was considered, but put aside for future work.

For Bendcast, an increase in the number of objects and cursor velocity had a negative effect on its suitability. A simplified version of the algorithm could be stated as $1 / (N \times V_C)$, where N is the number of objects in the cursor, and V_C 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 increased as the number of objects and cursor velocity increased, up to 36 objects. This is due to the fact that Expand was only designed to fit 36 objects into its virtual grid at any one time. Up to this number of objects, its suitability equates to: $(N / 36) + (V_C / 2)$. After 36 objects, its suitability would to decrease in a linear manner, reaching a nominal value at 40 objects. Beyond this point, another selection technique such as SQUAD [9] might be best. In our experience, it was essentially impossible to fit this many objects within the cursor, making this upper limit of Expand a non-issue.

As a result of our inclusion of these two selection techniques, Bendcast was 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.

4.4 User Feedback

To inform the user of which selection technique is currently active, a feedback mechanism was used. We designed a custom indicator icon, which was placed in the upper-right corner of the cursor. Each selection technique had its own icon, and it gave a hint as to how the technique would function. Bendcast featured a hand with a pointing finger with a laser emitting from it. Expand featured a 3 × 3 grid of colored blocks, which represent the virtual grid of objects. 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.

5 Summative Evaluation

To determine how well our algorithm faired against the two static techniques, we conducted a thorough user study across a variety of different scenarios. The study was influenced by a pilot study conducted previously.

5.1 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 entire experience for each participant took approximately 30 minutes, which included both a pre-questionnaire and post-questionnaire.

Our system setup featured a 50" HDTV, an Intel Core-i7 Laptop with an Nvidia GeForce GTX 560M GPU, and a Sony PlayStation 3 (PS3) featuring the Move camera and controller. The software used for development of our simulation was Unity 3.5, created by Unity Technologies. All Unity software was written in C#.

5.2 Experimental Task

Our participants were asked to perform selection tasks across 27 different scenarios that varied in number of objects, object velocity, and selection technique. For each scenario, a distinct object would be the desired target, and users were instructed to select it as quickly and efficiently as possible. Each scenario was performed five times, for a total of 135 selection events. The order of the scenarios was randomized to avoid any bias due to gained experience. Before starting the trials, the participant was given one minute to practice, which utilized the Auto-Select framework to expose them to Bendcast, Expand, and the Auto-Select techniques. For each scenario, they were given 2 seconds to observe the scene, locate the target, and determine 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.

5.3 Experimental Design and Procedure

We used a 3 × 3 × 3 within-subjects factorial design where the independent variables were selection technique (including auto-selection algorithm variant), scenario density, and object velocity. The selection techniques included Bendcast, Expand, and Auto-Select. The dependent variables were total selection completion time per scene and number of attempts required.

5.4 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 user's avatar stood where 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. 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 type of motion performed by each object was linear in nature, where each was moving at a constant rate of speed and fairly predictable. 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.

5.5 Experiment Results

Table 1: Results for (B)endcast, (E)xpand, and (A)utoselect

	Completion Time			Errors		
Density,Speed	B	E	A	B	E	A
100,2	0.87	2.27	1.65	0.08	0.21	0.07
100,4	1.06	2.48	1.87	0.17	0.03	0.10
100,6	1.48	3.14	2.34	0.29	0.07	0.13
200,2	1.05	2.64	1.88	0.21	0.05	0.09
200,4	1.40	2.46	2.44	0.47	0.00	0.23
200,6	1.70	3.27	2.82	0.53	0.07	0.32
300,2	1.21	2.69	2.50	0.36	0.02	0.07
300,4	1.78	3.23	2.57	0.77	0.08	0.21
300,6	2.45	3.55	3.32	0.91	0.16	0.34
Average	*1.44*	*2.86*	*2.38*	*0.42*	*0.08*	*0.17*

The average time per user per scenario for each technique is shown in Table 1. Based strictly on these times, Bendcast was the fastest in each scenario, with Auto-Select somewhere in between it and Expand. 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).

Error data is also presented in Table 1. 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 more errors than Auto-Select (t_{26} = 7.28, p < 0.01).

6 Discussion

As seen in Table 1, Bendcast took less time than Expand in all nine scenarios. Conversely, Expand always experienced fewer errors that Bendcast. Both of these results must be considered in a holistic manner to determine how to evaluate the effectiveness of the Auto-Select algorithms. In an ideal situation, the 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 tuning. There are so many factors that influence selection difficulty, and they can vary wildly with the interface.

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. It only makes sense that the results for Auto-Select were somewhere in between the other two techniques for both time and errors, since it chose from them. 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 may take a holistic approach to find the most balanced compromise in performance.

An important note to consider is that there was no penalty for an incorrect selection. As a result, there was no incentive 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 inherently 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.

Ultimately, there will always be underlying factors that prevent the performance from reaching the best of any possible technique. The act of switching techniques after the user has already started the mental process of performing a selection or while tracking a moving object will cause confusion and lower the user's confidence. Feedback mechanisms will not be effective in these cases, as the user is mentally involved in performing other motor tasks and is not focusing on the feedback at that point. Many other factors can play a role, and these need to be further researched.

7 Future Work And Conclusion

Our Auto-Selection technique and framework provides a foundation for future research in optimal selection technique assignment. The results show that it is a promising idea, and can provide an improved user experience. We understand that there is still room for improvement in several places. To better understand our results, we would like to perform a cognitive test before starting our next user study to better differentiate the experienced users from the inexperienced ones. Also, we would like to consider different selection techniques and suitability criterion, as well as testing the method of allowing selection techniques to access the entire scene without limitation.

Acknowledgements

This work is supported in part by NSF CAREER award IIS-0845921 and NSF awards IIS-0856045 and CCF-1012056.

References

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

[2] 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.

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

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

[5] O. Chapuis, J.-B. 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

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

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

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

[15] C. Wingrave and D. Bowman, "Baseline Factors for Raycasting Selection," *Proceedings of HCI International,* 2005.

[16] 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.