

# Automatic 3D Selection Technique Assignment Using Real-time Scenario Analysis

Jeffrey A. 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. Scene attributes such as object velocity, scene density, and user's cursor velocity can impact the user's ability to accurately select an object. Many 3D selection techniques have been explored, and are usually optimal for a specific set of conditions. As a result, software developers must compromise by choosing a single technique that works well on average, but is lacking in at least one scenario. We present a preliminary study that explores the feasibility of new auto-selection algorithms that automatically determines the most appropriate selection technique in real-time, thus leveraging the performance benefits of each technique. We evaluated the techniques across three levels of scene density and three levels of object velocity.

**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. With regard to 3D selection, much work has been done in developing new and efficient techniques to tackle many different types of scenarios [1]. It is generally the case that any single technique will perform well in certain scenarios, but not in others. Currently, a single technique must be put into place which is the best balance of performance across all scenarios, leading to compromise where it is inefficient.

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, but our work applies 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 determination algorithms. Our approach involves the design of a framework which allows many selection techniques to be considered by a high-level analyzer which determines the optimal technique at any given time.

---

\* e-mail: jcashion@knights.ucf.edu

+ e-mail: cwingrav@eeecs.ucf.edu

‡ e-mail: jjl@eeecs.ucf.edu

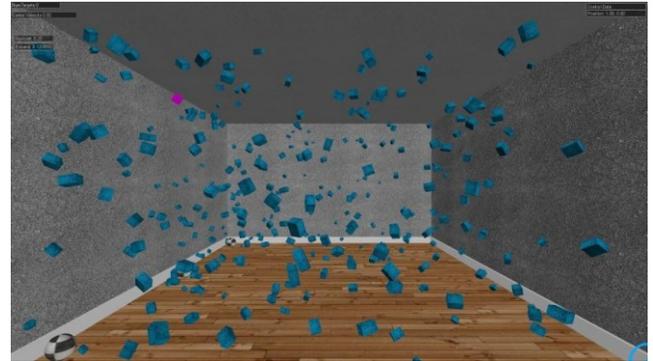


Figure 1: Selection Arena

## 2 RELATED WORK

One of the fundamental components to 3D interaction is selection [1]. It has been studied in great detail, and a large variety of selection techniques have been designed [2] [3]. 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 [4]. Some have even attempted to computationally define what a selection technique should be [6]. We explore this area further by computationally analyzing 3d selection techniques.

## 3 AUTOMATIC TECHNIQUE ASSIGNMENT

We developed an algorithm that can be used to determine which selection technique is best for any particular set of conditions. Specifically, 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. It is intended to be flexible and scalable, and developer-friendly.

### 3.1 Framework

The Analyzer is responsible for polling all of the available selection techniques and asking them what their suitability index is, given the provided conditions. Once all results have come back, the optimal technique is then chosen. To reduce rapid technique changing, a delay was inserted to prevent the user perceiving the current technique as toggling rapidly. For a technique to be considered by the analyzer, it must first register with it. This is done by calling the register method and passing in a reference to itself, the selection technique. It is required that the technique implement the "ISelectionTechnique" interface that we developed, so that the analyzer can agnostically interact with it. Whenever a selection attempt is made, the analyzer calls a "doSelection" method in the currently optimal technique, which then takes over control and does whatever it is designed to do.

### 3.2 Scene Analysis

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 three: the number of objects inside of the cursor, the average velocity of objects within the cursor, and the cursor velocity. Other factors that were not chosen but could be incorporated include average distance to objects inside cursor, average size of objects inside the cursor, level of occlusion, and more. We made our decision based on some analysis of which factors we felt would most significantly affect the difficulty of selection.

### 3.3 Auto-Select Algorithms

We developed two algorithms, which we call Auto-Select A and Auto-Select B. They were both assigned with choosing between the Raycast and Expand 3D selection techniques [2]. Both algorithms computed their suitability index based on cursor velocity and the number of objects within the cursor. Algorithm B was based on A, with the inclusion of average object velocity. The reason for developing two algorithms was to determine if we could observe a difference in performance. This would encourage future work in other algorithms that may see even better results. In addition, it might be the case that a particular selection technique is more suitable to analyzing attributes which differ from those that we chose.

### 3.4 User Feedback

The algorithm that performs the auto-selection is very important, but another key component to the entire framework is how to inform the user that such a change of technique is taking place. The method that we implemented was the design of 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.

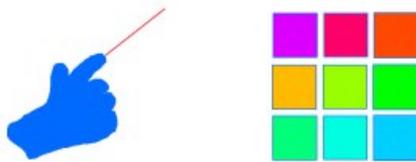


Figure 2: Feedback Indicators: Raycast (L) and Expand (R)

## 4 SUMMATIVE EVALUATION

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 entire experience for each participant took approximately 20 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. The PlayStation 3 was utilized for its Move.Me SDK [5].

Our participants were asked to perform selection tasks under 36 different scenarios that varied in number of objects and object velocity. The user would select a single pre-determined object which was colored differently to indicate its importance. The scenario order was randomized, as well as the selection technique assigned.

We used a  $4 \times 9$  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 quantity of objects was 100, 200, or 300. The average object velocity was 2, 4, or 6 meters per second.

## 5 EXPERIMENT RESULTS AND DISCUSSION

Raycast performed worse than Expand in all nine cases, both in terms of time and errors. Both of our Auto-Select algorithms performed very similarly, with no statistical difference between the two. From this, we made several observations. Being "in the zone" is a factor that we believe plays a role in performance [7]. This is essentially a user focusing so much on what they are doing that they are not cognizant of the feedback indicator, thus not utilizing important information in performing a selection. 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. The quality of the feedback mechanism also plays a large role in the user's ability to understand which technique is currently active. 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.

## 6 FUTURE WORK AND CONCLUSION

We have presented a novel technique for determining which selection technique is optimal at any given moment. 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. The study we performed shows that there is definitely a potential for more advanced auto-selection algorithms, which can be custom tailored by the adopters. We have made a good initial effort in analyzing this solution, and believe that with further work and user studies, the techniques outlined could be improved.

### ACKNOWLEDGEMENTS

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

### REFERENCES

- [1] D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev, 3D User Interfaces: Theory and Practice, Addison-Wesley, 2004.
- [2] 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.
- [3] 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.
- [4] I. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction," *Human Computer Interaction*, vol. 7, pp. 91-139, 1992.
- [5] "Sony Move.Me SDK," 2012. [Online]. Available: <https://us.playstation.com/ps3/playstation-move/move-me/>.
- [6] A. Steed, "Towards a General Model for Selection in Virtual Environments," *IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 103-110, 2006.
- [7] 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.