Dynamic Adaptation of 3D Selection Techniques for Suitability Across Diverse Scenarios

Jeffrey Cashion*
University of Central Florida

Joseph J. LaViola, Jr.‡
University of Central Florida

ABSTRACT

We performed a user study that measured the effectiveness of our new 3D selection technique, Scope, which dynamically adapts to the environment by altering its activation area and visual appearance with relation to cursor velocity. Users tested our new technique against existing techniques Raycast, Bendcast, and Hook across a variety of different 3D scenarios which featured three different levels of object density and three different levels of object velocity. Our two dependent variables were completion time and total attempts per scenario. Users also completed a post-questionnaire which yielded qualitative insights on their experience. Our study shows that Bendcast, Scope, and Hook all performed similarly across all scenarios, yet were all significantly faster and less error-prone than Raycast. Despite this similar performance, users strongly favored Scope over the other three techniques, and over the second most preferred technique nearly two to one.

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 rapid increase in graphical processing power has made 3D graphics much more accessible to common devices. Scenes can be much richer and more detailed, with a higher number of objects and richer detail. With 3D visuals being the primary output of such software, one of the primary inputs is user selection. In video games, a user may select something as simple as a menu option, or something as complex as a small moving object hidden in foliage. Scientific applications such as medical simulators or bio-molecular visualization also rely heavily on precise and accurate selection. In any case, it is important that the selection techniques utilized best match the expected environment and its conditions. With such importance placed on selection, it is no surprise that much work has been done to reduce selection time, reduce errors, and increase overall usability [6] [3]. Comprehensive research has been done on categorizing and grouping techniques by common features and characteristics [1]. A primary characteristic of a technique is whether it operates statically or dynamically. Our work focuses on further exploring the area of dynamic selection techniques. Our goal was to develop a technique that would adapt to user input and dynamically change how it operated. We explored the existing field of techniques and identified a few that had features which could be worked with.

*jcashion@knights.ucf.edu
‡jjl@eecs.ucf.edu

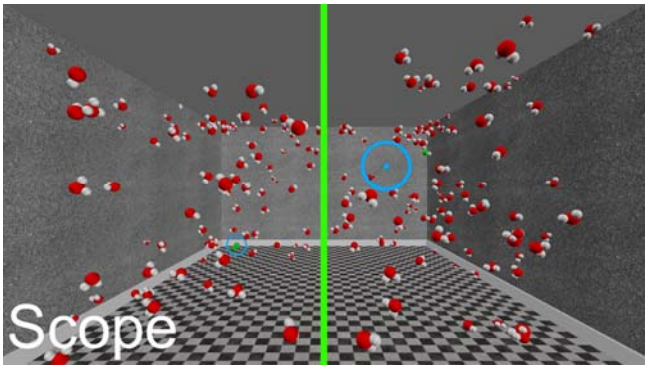

Figure 1: Scope, slow user input (left), fast user input (right)

2 Related Work

The primary two techniques that form the baseline of 3D selection are Raycast and SpotLight [1]. Raycast features a single ray that is projected into the scene, while SpotLight projects a cone-like shape into the 3D world instead of a single ray. For DynaSpot [4], the cursor acts as a circle that varies in diameter; as small as a point when the cursor is motionless, and a large circle when the cursor is moving quickly. Bendcast is a technique that bends to the target nearest to the center of the closed static cursor [2]. The Hook technique was designed to operate by using a scoring system over time [6], in much the same way that IntenSelect operates [5]. The primary difference is that Hook computes the distance from the cursor to each object, and derives a score based on that measurement. Expand is an example of a selection technique that features some dynamic characteristics [2]. The technique uses a two-step process that arranges selected objects into a virtual grid, giving the user a better opportunity to select.

3 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. To achieve this, we utilized ideas from different selection techniques and tried to make them all work well together. 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. We then branched out and made some modifications and introduced some new ideas.

3.1 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 [2]. The speed-dependent behavior of DynaSpot was shown to give it an 18% performance advantage over Raycast. Due to this, we built off of that idea 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 diameter of the cursor (in pixels) is

$$\text{SPOTSIZE} = \textit{Clamp}(\text{CURSORVELOCITY} \times \alpha, S_{MIN}, S_{MAX})$$

where α adjusts the sensitivity to the user input, and S_{MIN} and S_{MAX} are the minimum and maximum sizes that the spot can be, respectively. 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 0.8 seconds. From this, the distance and time was measured between the points to evaluate what the average velocity was over that range. This benefited us by providing a natural ramping delay in the size of our spot.

3.2 Nearest-Object Determination

Our approach for computing the nearest object is 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 to indicate its victory.

4 Summative Evaluation

We conducted a user study with 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. Each participant took approximately 30 minutes to complete the study, and was compensated $10 for their time. This time included the completion of both a pre- and post-questionnaire. Our test configuration included an Intel Core-i7 laptop with 16GB of RAM, a GeForce GTX 560M, and a 55" 1080p display. We utilized a Sony Move controller, and Unity 4.2 software.

4.1 Experimental Design and Procedure

We utilized a 4 x 3 x 3 within-subjects factorial design, with the 4 selection techniques, 3 scene densities (low, medium, high), and object velocity (still, moderate, 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. The timer would start when the user was permitted to begin the selection task, and would end upon selection of the correct object. Each participant was informed that they would be performing a series of selection tasks. Their goal in each task was to select a specific object, which was marked green. The order of the tasks was randomized for each participant, and each participant was given sixty seconds to practice the selection techniques at first.

4.2 Experiment Results and Discussion

Each participant completed 5 runs of the experiment, with each run containing 36 scenarios. 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.

We found significant differences between the techniques for average total time per technique ($F_{3,27} = 76.2$, $p < 0.001$). With individual t-tests, Raycast was significantly slower than Hook ($t_{26} = 9.665$, $p < 0.0083$), Scope ($t_{26} = 9.412$, $p < 0.01$), and Bendcast ($t_{26} = 8.796$, $p < 0.0125$). Significant differences were found in average errors per technique ($F_{3,27} = 269.1$, $p < 0.001$), 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 ($t_{26} = 2.876$, $p < 0.025$). A Chi-squared test on the participant rankings for preferred technique revealed that Scope was most preferred ($\chi^2_3(N=27) = 9$, $p < 0.05$).

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. We were pleased to see that users did strongly favor our new technique, Scope, over the competition. It is interesting to consider both the similarity in performance and the stark difference in user preference. The users acknowledged that they considered the usability, speed, and accuracy to be roughly the same as Bendcast and Hook, and the quantitative data showed that they did in fact perform approximately as well as each other.

5 Future Work And Conclusion

Our attempt at creating a dynamic selection technique showed that there is potential in designing new techniques that will also improve performance. The primary areas that we would like to focus our efforts include best-target determination, visual interaction, and scoring algorithms. We believe that with continued efforts, we will develop a more thorough understanding of these considerations, which will ultimately lead to the design of more advanced techniques.

Acknowledgements

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

References

[1] 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.

[2] 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.

[3] 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,* pp. 107-110, 2013.

[4] 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.

[5] 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.

[6] M. Ortega, "Hook: Heuristics for Selecting 3D Moving Objects in Dense Target Environments," *3D User Interfaces (3DUI), 2013 IEEE Symposium on,* pp. 119-122, 2013.