Object Selection Techniques in 3D Interactions

Can Li

Abstract

Object Selection is a primary technique used in 3D interaction. Although many techniques has been implemented, few of them can support selection objects in dense area or under occlusion. This paper introduces several design guidelines of selection techniques. In order to support those guidelines, two techniques -- bubble cursor and depth ray, are introduced. Moreover, considering the complexity and unpredictability in 3D environment, some auxiliary techniques such as feature analysis and extraction and tour path recommendation could be done before selection to achieve better performance. Finally, a framework containing all those techniques is proposed for object selection in 3D interactions.

1. Introduction

In recent years, virtual reality (VR) become intensively popular not only in academic research but also in many industry applications. It is even expected as the next revolution in the foreseen future. One of the primary techniques which must be supported in any interactive 3D application is object selection. Even though it's a trivial task in computer graphics by using mouse, it become difficult since the targets will have 3D coordinates, which the user must somehow specify. As such, it is critical for VR researchers to develop new selection techniques for VR environment.

In fact, numerous techniques have been developed in VR for object selection, which has been repeatedly defined as a fundamental task. Hence, one of the most common techniques is hand extension techniques [1], which applies mapping between hand or handheld input device and the 3D coordinates of a virtual cursor. The other technique is ray-casting, which casts a ray into the 3D scene and calculate intersections with the target. However, those techniques don't fully consider many factors which may lead to failure of object selection. In this paper, we mainly discuss three factors.

- Density of targets: This can significantly affect the performance of the selection technique derived from ray-casting. As we know, the first intersected object would be selected in this kind of algorithm. Thus, it is very difficult to select those objects behind dense areas
- Visibility of targets: In 3D, the target may be occluded by other objects in the scene, in which case, the user is not be able to select the target and is required to rotate the scene or switch to other viewports, which are time-consuming.
- Auxiliary techniques: No matter what techniques adopted, the two factors above are inevitable. Some auxiliary techniques can be applied to help object selection.

This paper discuss different object selection techniques corresponding to those factors. Based on those, a general framework is proposed for choosing proper object selection techniques. Also, some user study cases are discussed to support the effectiveness of those techniques.

2. Related Work

Object selection is defined as one of the four basic interactions (along with navigation, manipulation and data input) in 3D virtual environment [1][2]. A lot of research have been conducted on this critical topic.

The earliest implementation of object selection using ray-casting is Liang and Green [3], called “laser gun” ray-casting technique. For this technique, a ray is casted from the user’s hand with the control of ray origin and trajectory. Then they proposed a technique called “spotlight selection”, where users emit conic selection area to overcome difficulty of selecting small and distant objects. Based on this, several research conducted improvement, such as aperture based selection [4] and 2D image plane selection [5]. However, those techniques cannot work accurately for objects behind dense areas.

Another alternative to the ray-casting is hand extension technique, where the user control X, Y ,Z coordinates of a 3D cursor [1][5]. Poupyrev et al. [6] extends the range which the cursor can cover by using nonlinear mappings between user’s hand and 3D cursor. One problem of those type of techniques is lack of efficiency due to mapping calculation. Zhai et al. [7] introduced a semi-transparent 3D volume cursor to to increase performance. However, this technique has difficulty when interacting with dense target environments due to multiple objects may fall into the boundary of the volume.

Other than the research field of VR, flow visualization research have some inspiring works on object selection. Tao et al [8] proposed an approach which extracts shape invariant features and encode those objects, which are streamlines. Then user can flexibly query and select the target by pre-defined code. Ma [9] proposed a novel framework which can select viewpoint and generate tour to explore the target objects. Those technique could be auxiliary ways for object selection in VR field.

3. Selection Techniques

3.1 Design Guidelines

Vanacken et al. [10] proposed six design guidelines of object selection techniques considering the factors mentioned above:

- Allow for fast selections.
- Allow for accurate selections.
- Be easy to understand and use.
- Produce low levels of fatigue.
- Satisfy the above for sparse and dense target environments.
- Support selections for both visible and occluded targets.

Since every 3D environment has its unique property, directly selecting on objects may not be necessary under different circumstances. One more guideline can be considered:

- Use auxiliary technique to further improve performance of object selection

3.2 3D Bubble Cursor

Vanacken et al. [10] first apply those guidelines to the hand extension technique. They use the bubble cursor to select objects. The bubble cursor is able to resize dynamically such that only the closest target falls within its boundaries, which alleviate the problem of containing multiple objects in the volume cursor. The bubble cursor would be rendered as a gray semi-transparent sphere as shown in Figure 1(a). When the real bubble is closing or leaving the object, a second semi-transparent sphere is rendered around the captured target, which is always appears to be completely contained by the cursor, as shown in Figure 1(b). Besides, the selected object is colored in yellow and a cross is added inside of the bubble to mark its center. The render algorithm to decide the size of the bubble is described in Grossman and Balakrishnan [11].

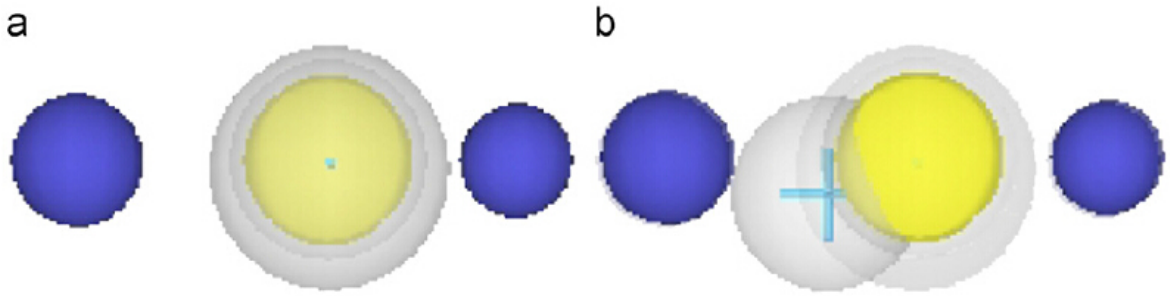

Figure 1 (a) The 3D bubble is rendered as a gray semi-transparent sphere; (b) The second bubble is rendered

Based on the design guidelines, the technique should support efficient selection in dense 3D environment as well as on occluded objects. In order to overcome such occlusions, the bubble cursor is given with the magic lens capability as shown in Figure 2(a), which allows the objects in its vicinity become semi-transparent by calculating the distance between the bubble cursor and each target, measured on the 2D image viewing plane. Thus, the user is able to observe the occluded the object when the bubble approach it, assuming they know its general location, as shown in Figure 2(b).

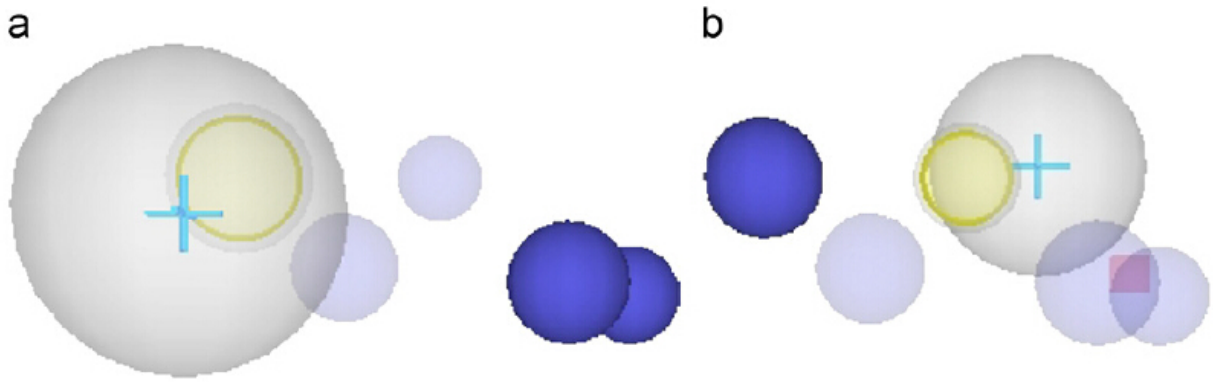

Figure 2 (a) Objects become semi-transparent when close to the bubble cursor; (b) Occluded objects become visible when near the bubble cursor.

It is worth to note that this technique would be more fit for the circumstance where the user has general understanding of the location of their targets. If the search of targets is needed, global methods such as rotating the scene or switching the viewport would be more appropriate.

3.3 Depth Ray

Similar to the last section where the guidelines is applied with the hand extension object selection technique design, Vanacken et al. [10] introduce the guidelines to ray-casting technique. As it's discussed before, ray-casting technique may select multiple objects due to several intersections along the ray.

Grossman and Balakrishnan [12] proposed the depth ray that can solve this ambiguity. The user controls a depth marker, which moves along with the hand. When hands go forward or backward, this depth marker would be aware of the change of the depth in 3D scene and do the same manner. The object intersected by the ray cursor, which is closest to the depth marker, would be selected. Like the bubble cursor, the selected target is highlighted yellow, as shown in Figure 3.

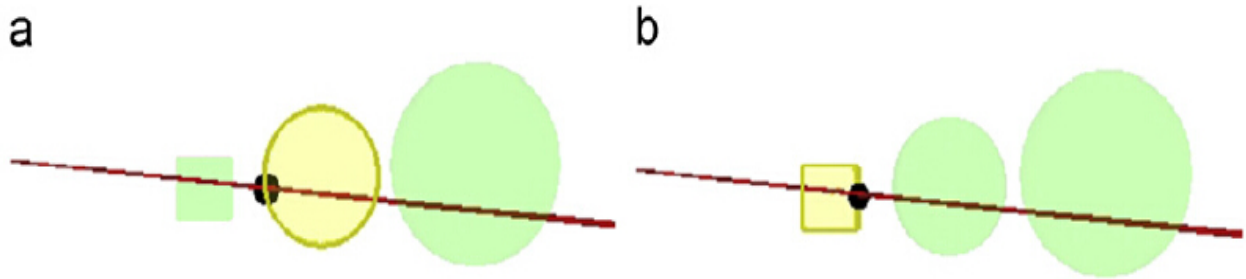

Figure 3 (a) The object closest to depth marker is selected; (b) Depth marker can be controlled by moving hand forward or backward.

In order to select the occluded targets, the similar transparency function used in bubble cursor can be introduced. When ray cursor is close to some objects within a specific distance, it translates the objects to semi-transparent.

3.4 Auxiliary Techniques

The above two techniques can directly select the targets in 3D environment. They can mostly satisfy the design guidelines. However, they have some drawbacks. First of all, they are all local techniques, which would not take global environment into count. For example, if the viewport is extremely occluded originally, in this case, even if objects near cursor can be semi-transparent, it is very difficult to select the target. Secondly, they are based on the fact that the users may have a pre-understanding of the whole 3D environment. If it is completely new to the user or amount of objects are too big, then the user needs to search 3D environment before selection. Hence, two auxiliary techniques are introduced below to handle some particular need during the object selection.
Moreover, 3D environment is way more complicated that we can predict so that sometimes the occlusion and dense area are inevitable. Even if those techniques above can select targets under occlusion or dense environment, the accurate would not be good enough. Thus, auxiliary techniques could be applied to achieve better performance of object selection.

3.4.1 Flow String

In flow visualization research, Tao et al. [8] presented a novel approach to encode the objects in 3D environment by analyzing their shape similarity. Specifically, this paper first resamples streamlines by considering their local feature scales. They then classify resampled points along streamlines based on the shape similarity around their local neighborhoods. They encode each streamline into a string of well-selected shape characters, from which meaningful words for querying and retrieval are constructed. The result is shown in Figure 4.

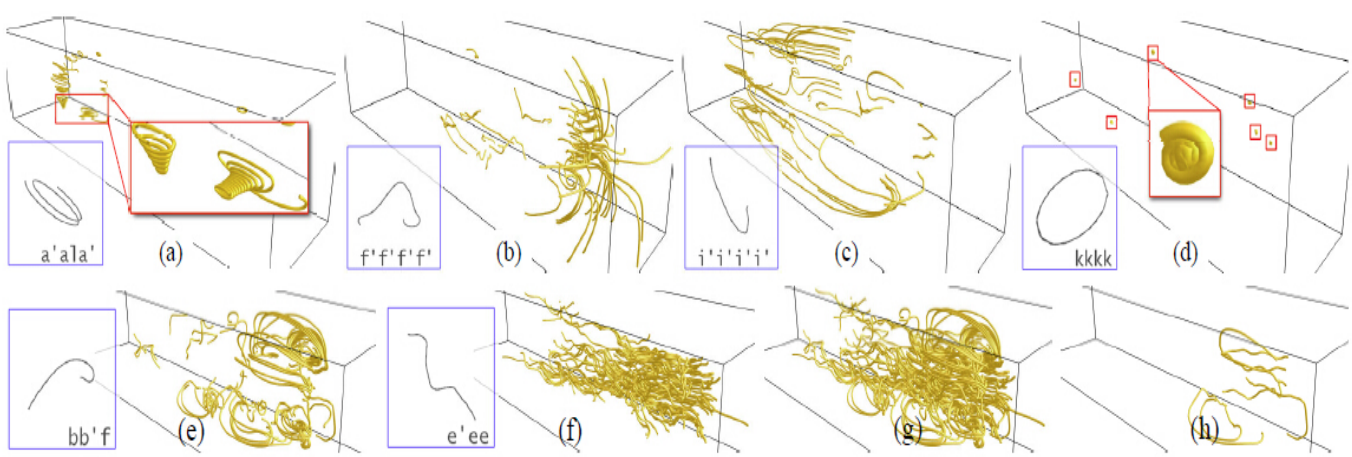

Figure 4 Flow string case study of plume dataset. (a) to (h) show streamlines matched by different words

Although this technique is applied in flow visualization, it inspire an indirect interaction that may be used in VR -- encoding the potential objects for easily searching them. In some cases,

users are only interested in selecting objects with particular features. Thus, we can first perform analysis on 3D environment to encode those objects based on their features. Then the user can filter the potential targets by figuring out their code, which possibly make 3D environment more sparse. In the end, the user may adopt hand extension or ray-casting technique to select the target.

3.4.2 Flow Tour

In flow visualization field, Ma et al. [9] proposed a novel framework that generates an automatic guide for exploring internal flow features. Specifically, their algorithm first detects objects that would be observed potentially. Then the best viewpoints would be found based on the occlusion and the feature of the objects. Finally, a tour connected those viewpoints in a smooth and efficient manner is generated for exploring the whole 3D environment. The result is shown in Figure 5.

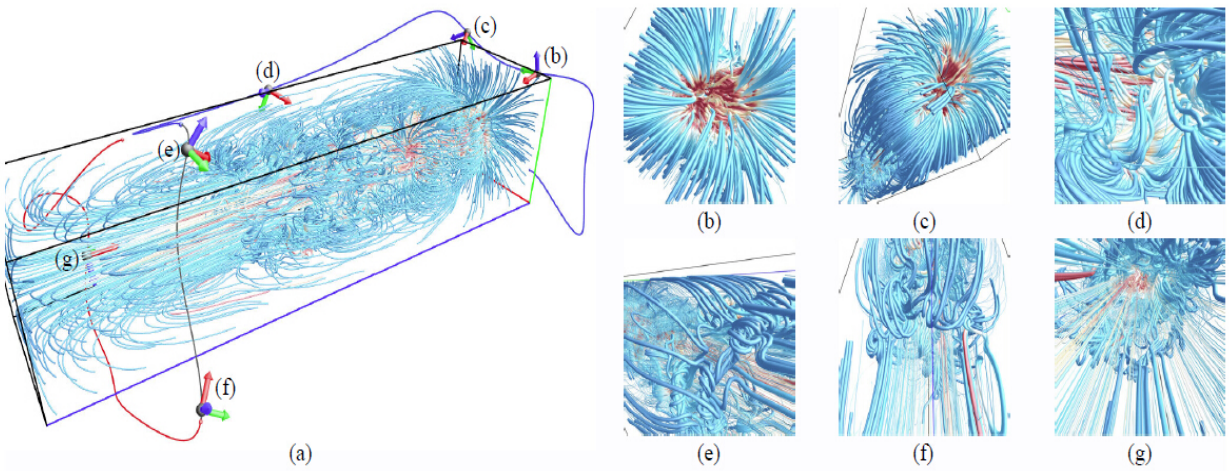

Figure 5 Flow tour for plume dataset. (a) is the whole tour generated by this approach; (b) to (g) show respective views from six viewpoints along the tour path as marked in (a)

The most inspiring parts in this paper are viewpoint selection and tour generation. It would be much easier for user to select objects if the initial viewpoint avoid occlusion and focus on potential objects. Also, if there are several objects need to be selected, a recommended tour path can be displayed, which may lead user to select targets in better viewpoints.

3.5 Framework of Object Selection

From above, in some cases it might be more effective to do some preprocessing than directly let user select targets without any instruction in the 3D environment, even if the selection techniques are state-of-the-art. Hence, we may consider the object selection process as a framework in Figure 1 containing several steps as following:

- Preprocessing
    - Analyze the 3D environment and the purpose of the selection behavior conducted by users
    - Extract features that users desire and eliminate unrelated objects
    - Provide indirect way to retrieve objects by encoding them
- Applying Selection Techniques
    - Bubble Cursor
    - Depth Ray
- Selection Recommendation
    - Recommend optimal path for object selection
    - Highlight potential targets

In the preprocessing, some auxiliary techniques could be applied to 3D environment to extract the desired feature, which could help elimination of unrelated objects to reduce occlusion. Also, based on the feature, encoding the objects would be possible to provide another way of object retrieval.

Then users can actually select the object using bubble cursor or depth ray. During this, potential objects could be highlighted as a instruction and recommending path could be displayed for users to select next target.

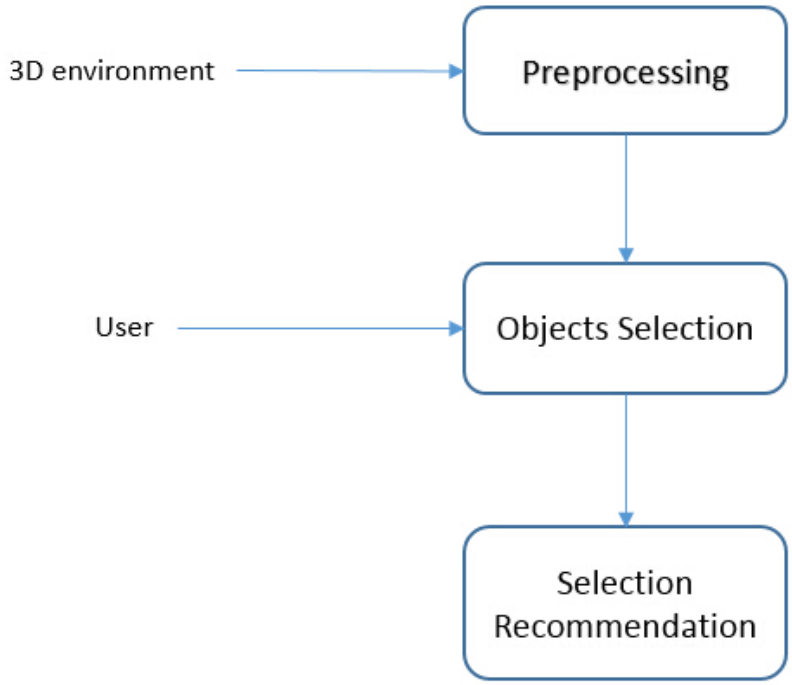

Figure 6 Framework of Object Selection

4. Conclusion

In this paper, several object selection techniques are introduced, including bubble cursor and depth ray. Those techniques are able to overcome the difficulty of object selection under dense area and occlusion. Moreover, other techniques such as flow string and flow tour used in flow visualization research inspire that some preprocessing could be done before object selection to achieve better performance. Finally, a framework combining all those techniques is proposed. In the future, this framework can be implemented and tested to prove its validity.

Reference

[1] Mine, M., 1995. Virtual environment interaction techniques. UNC Chapel Hill Computer Science Technical Report TR95-018.

[2] Bowman, D.A., Kruijff, E., LaViola, J.J., Poupyrev, I., 2004. 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Reading, MA.

[3] Liang, J., Green, M., 1994. JDCAD: a highly interactive 3D modeling system. Computers and Graphics 18, 499–506.

[4] Forsberg, A., Herndon, K., Zeleznik, R., 1996. Aperture based selection for immersive virtual environments. In: ACM UIST Symposium on User Interface Software and Technology, Seattle, Washington, USA. ACM Press, New York, pp. 95–96.

[5] Hinckley, K., Pausch, R., Goble, J.C., Kassell, N., 1994. A survey of design issues in spatial input. In: ACM UIST Symposium on User Interface Software and Technology. ACM, New York, NY, pp. 213–222.

[6] Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T., 1996. The go–go interaction technique: non-linear mapping for direct manipulation in VR. In: ACM UIST Symposium on User Interface Software and Technology, pp. 79–80.

[7] Zhai, S., Buxton, W., Milgram, P., 1994. The ‘‘Silk Cursor’’: investigating transparency for 3D target acquisition. In: ACM CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, pp. 459–464.

[8] Tao, J., Wang, C., Shene, C., 2014. FlowString: Partial Streamline Matching Using Shape Invariant Similarity Measure for Exploratory Flow Visualization. In: Proceedings of the 2014 IEEE Pacific Visualization Symposium (PACIFICVIS '14). IEEE Computer Society, Washington, DC, USA, 9-16.

[9] Ma, J., Walker J., Wang C., Kuhl S., Shene C., 2014. FlowTour: An Automatic Guide for Exploring Internal Flow Features. In Proceedings of the 2014 IEEE Pacific Visualization Symposium (PACIFICVIS '14). IEEE Computer Society, Washington, DC, USA, 25-32.

[10] Vanacken, L., Grossman, T., Coninx K., 2008. Multimodal selection techniques for dense and occluded 3D virtual environments. In: Human-Computer Studies 67 (2009). Autodesk Research, Toronto, pp. 237-255.

[11] Grossman, T., Balakrishnan, R., 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: ACM CHI Conference on Human Factors in Computing Systems, Portland, Oregon, USA. ACM Press, New York, pp. 281–290.

[12] Grossman, T., Balakrishnan, R., 2006. The design and evaluation of selection techniques for 3D volumetric displays. In: ACM UIST Symposium on User Interface Software and Technology. ACM Press, New York, pp. 3–12.