

Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments

Lode Vanacken*

Tovi Grossman†

Karin Coninx‡

*Hasselt University, Expertise Centre for Digital Media and transnationale Universiteit Limburg
Wetenschapspark 2, B-3590 Diepenbeek, Belgium

†Dynamic Graphics Project
Department of Computer Science, University of Toronto
www.dgp.toronto.edu

ABSTRACT

Object selection is a primary interaction technique which must be supported by any interactive three-dimensional virtual reality application. Although numerous techniques exist, few have been designed to support the selection of objects in dense target environments, or the selection of objects which are occluded from the user's viewpoint. There is, thus, a limited understanding on how these important factors will affect selection performance. In this paper, we present a set of design guidelines and strategies to aid the development of selection techniques which can compensate for environment density and target visibility. Based on these guidelines, we present two techniques, the depth ray and the 3D bubble cursor, both augmented to allow for the selection of fully occluded targets. In a formal experiment, we evaluate the relative performance of these techniques, varying both the environment density and target visibility. The results found that both of these techniques outperformed a baseline point cursor technique, with the depth ray performing best overall.

CR Categories and Subject Descriptors: I3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques.
Additional Keywords: Virtual reality, 3D interaction, selection.

1 INTRODUCTION

In recent years, three-dimensional (3D) display technologies, such as immersive Virtual Reality (VR) systems [5], or non-immersive fish-tank VR systems using LCD shutter stereo-glasses [27], have significantly improved in display quality. Experimental evaluations have also shown that these displays can improve the user's ability to perceive virtual 3D scenes [28], making them a potentially beneficial alternative for 3D applications.

One of the primary techniques in any interactive application which must be supported is object selection. Within the realm of 3D virtual environments, selection has been repeatedly identified as one of the fundamental tasks [4, 18]. However, when selecting objects in a 3D environment, the standard 2D mouse metaphor breaks down, as the targets will have 3D coordinates, which the user must somehow specify. As such, it is important for VR researchers to consider new selection techniques, specifically designed for 3D environments.

Indeed, research in VR environments has introduced numerous techniques for object selection. Most commonly seen are hand extension techniques [18], for which the 3D coordinates of the

hand or handheld input device are mapped to the 3D coordinates of a virtual cursor, and ray casting techniques [15], for which a virtual ray is cast into the scene, and made to intersect targets of interest. Despite the numerous designs and evaluations of such techniques, there are a number of important factors which remain to be fully understood, two of which we focus on in this paper.

The first is the environment density of targets. This can greatly affect the performance of a selection technique, particularly in a 3D environment. For example, with ray casting techniques, multiple targets can be intersected either along the length of the ray, or in some cases, within its conic selection area [15]. This results in an ambiguity of the user's intended target. Recently, researchers have attempted to address this problem of ambiguity with new selection techniques [13, 19, 29], but the effect of target density on such techniques remains to be explored.

The second factor which we investigate in this paper is the visibility of the goal target. While, in many cases, users' intended targets are visible, there may be cases where the intended targets are occluded from their view by another object in the scene. Generally, users are required to either rotate the scene or to switch to a different viewing mode, to allow them to select their desired target. These extra steps may be time consuming, requiring the addition of modes and buttons to the interface. It is, therefore, of interest to develop techniques which allow users to seamlessly select targets which are occluded from their view, and to understand how the visibility of the target affects selection performance.

In the following, we present two candidate selection techniques, the 3D bubble cursor and the depth ray, which allow for efficient selection of targets in both sparse and dense environments. Furthermore, we augment these techniques to also allow for the selection of targets which are not visible from the user's viewpoint. After discussing the design of these techniques, we present a formal experiment, comparing our proposed techniques to a standard point cursor, also augmented to allow for the selection of occluded targets. The results show that both the depth ray and the 3D bubble cursor outperform the point cursor in all conditions, with the depth ray performing best overall. Furthermore, both techniques adequately allowed users to select targets which were not visible from their initial viewpoint.

2 RELATED WORK

Researchers in 3D virtual environments have often categorized selection as one of the four basic interactions (along with navigation, manipulation, and data input) [4, 18]. Selection allows users to specify an object, with which they wish to manipulate or interact. Because it is such a critical task, there has been a wide variety of research looking into various techniques for supporting it. In this section, we first provide a review of selection techniques for 3D environments. We then discuss selection techniques which have been studied under dense target environments, followed by a discussion of techniques for overcoming target occlusions.

*lode.vanacken@uhasselt.be

†tovi@dgp.toronto.edu

‡karin.coninx@uhasselt.be

2.1 3D Selection Techniques

One of the earliest implementations of selection for 3D environments was done by Liang and Green [15]. They implemented a ray casting selection technique that they called “laser gun” selection. With this technique, a ray is emitted from the user’s hand, so the user has control over the origin and trajectory of the ray, much like using a physical laser pointer. One observed problem with this technique was that it was difficult to select distant and small objects due to the angular accuracy which was required. To overcome this, they introduced a technique called “spotlight selection” where, instead of emitting a ray, the user emits a conic selection area, originating from the user’s hand. Since this original work, there have been a number of iterations on the ray casting metaphor in the 3D research community, such as aperture based selection [9] and 2D image plane selection [20]. However, a general problem with the ray casting metaphor is that in dense target environments, multiple targets can be intersected by the ray, or fall within the conic selection area, creating an ambiguity of the intended target. Existing strategies to address this ambiguity are discussed in the next section.

The most common alternative to the ray casting metaphor is the hand extension metaphor. This is a more direct method of interaction, for which the user controls the X, Y, and Z coordinates of a 3D cursor with a mapping from the 3D position of the hand [14, 18, 21]. Mine [18] states that in local interactions, a direct mapping from the user’s hand to a 3D “virtual cursor or drone” could be used to select an object. For distant objects, the go-go technique explores the use of nonlinear mappings between the user’s hand and 3D cursor, extending the range which the cursor can cover [21]. One drawback of the hand extension metaphor is that the selections are constrained by three dimensions, resulting in longer selection times. This has been confirmed in a number of studies [3, 10, 13, 22]. As an alternative to a simple 3D point cursor, Zhai et al. [30] developed the silk cursor, which is a semi-transparent 3D volume cursor. Using a volume increases the activation area of the cursor, and it was shown to increase performance in a 3D target tracking task. While a volume cursor could reduce target acquisition times, it, once again, produces difficulty when interacting in dense target environments, as multiple targets may fall within the boundaries of the cursor’s volume. We now discuss techniques which specifically address selection for dense target environments.

2.2 Selection Techniques for Dense Environments

For ray casting techniques, generally only the first intersected object will be selected, even though the ray can intersect multiple objects simultaneously. Under this implementation, it may be difficult or even impossible to select objects that are further away, depending on the density of the target environment. To address this shortcoming, the following techniques have been proposed.

Liang and Green [15] developed a metric for the spotlight selection to determine which object would be selected when multiple targets were captured, based on the distance between the target to the apex and central axis of the cone. An interesting extension to spotlight selection is Shadow Cone Selection [24], which selects targets by sweeping out an area with a cone selection cursor. Other metrics have also been proposed [25], but it is unclear how well they will work in dense target environments, as they have not been formally evaluated.

An alternative to defining these predictive metrics is to provide an explicit mechanism for users to specify their intended target among all those intersected [14]. Grossman et al. [11] used forwards and backwards hand movements to cycle through

intersected objects, however, limited visual feedback was provided to the user. Olwal et al. [19] describe the flexible pointer, which allows users to bend the cursor to avoid intersecting other targets. A drawback of this technique is that it requires two 6-dof devices to control the cursor. Another technique requiring two input devices is iSith [29], where two rays are simultaneously controlled, with their intersection being used to define a target.

In a recent study, Grossman et al. [13] discussed the design of several new ray casting techniques which allow for multiple target disambiguation. This is one of the few formal studies evaluating such techniques. Of their tested techniques, they found the depth ray to be most successful. The depth ray augments the ray cursor with a depth marker, which can be moved forwards and backwards along the length of the ray, with similar movements of the hand. The intersected target closest to this depth marker is the one which can be selected. Although the study used a dense environment, the environment was constant throughout the experiment, so they were not able to study the effect of the environment density on the technique. While this study was performed on a 3D volumetric display, the results can be applied to other implementations of 3D virtual environments.

There has also been work in selecting targets in dense two-dimensional environments. Most notably, the bubble cursor [12] is an area cursor that dynamically changes its size to always capture only the closest target. The technique was studied in environments of varied density, and found to provide efficient selections in sparse and dense environments. Such a technique could be an interesting alternative to a static 3D volume cursor, which is an idea we will explore in this paper.

2.3 Overcoming Target Occlusion

Some of the above techniques for dense environments explore the scenario of multiple objects occluding the ray cursor’s approach to a goal target. However, none explore the issue of a target being completely occluded from the user’s viewpoint. Our literature review found limited research on this topic. In 3D desktop applications, users generally rotate the scene so that their target of interest becomes visible. Another approach is to switch to a viewing mode, such as wireframe rendering, so that all targets become visible. Some more specialized viewing modes have also been studied in augmented reality environments [16]. Other techniques to reduce occlusions, such as interactively distorting the space [6, 7, 17], or morphing the viewing projection [8], have also been explored. However, these techniques are generally independent from the selection mechanisms.

Some relevant research in the 2D realm are the tumble and splatter techniques [23]. These techniques allow users to first spread out layers of objects which are occluding each other in a 2D environment so that they are all visible. The user can then select and perform interactions with any of the objects.

In summary, there is vast research on selection techniques for 3D environments. However, less research has focused on supporting selection in dense target environments, and even less on the selection of objects which are fully occluded. While some promising techniques do exist, to date there has not been an exploration of how these techniques are affected by the environment density and target visibility.

3 DESIGN GUIDELINES AND STRATEGIES

Before introducing the techniques which we present and evaluate in this paper, we first discuss some high-level design guidelines which we had for these techniques. In this section, we present these design guidelines, followed by our proposed design strategies for satisfying these guidelines.

3.1 Design Guidelines

Our literature review of the previous work on selection techniques in 3D environments revealed that the environment density and visibility of the goal target are factors which are not well understood. As such, it is our goal to design and evaluate techniques which can adequately account for these two variables. In addition to some standard design guidelines for 3D selection techniques, this gives us the following six design guidelines:

- 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

While previous work will guide us in satisfying the first four design guidelines, we propose two design strategies for satisfying the last two. We now discuss these two design strategies in detail.

3.2 Increased and Unambiguous Activation Areas

For a selection technique to be appropriate for use in an actual interface, it should support efficient selections in both sparse and dense environments. However, supporting efficient selections for both hand extension and ray casting techniques is a difficult task, due to the constraints imposed by the task. For hand extension techniques, the added third dimension imposes both a physical and visual constraint which can slow down selection times [3, 10]. For ray casting techniques, small angular changes in the input device can result in large movements of the ray [15]. Our strategy for reducing the effects of these constraints will be to increase the activation areas of the selection techniques. Established methods for this include: using a selection volume for hand extension techniques and emitting a cone for ray casting techniques [15, 30].

Unfortunately, increasing the cursor’s activation area means multiple targets can be captured simultaneously, introducing an undesirable ambiguity. This can be especially problematic in dense target environments. We, thus, seek 3D selection techniques which provide a mechanism for disambiguating between potential targets of interest. One strategy would be to use a dynamic activation area [12]. Another alternative is to provide a mechanism for disambiguating between multiple captured targets [14]. Such techniques have a better chance of being efficient in both sparse and dense target environments.

3.3 Integrated Visual Enhancements for Occluded Targets

It is often the case in 3D environments that users need to select objects which are obscured from their viewpoint. This is generally the case when the target of interest lies behind another object in the scene and is thus occluded. If the environment is densely populated with targets, the chance of such an occlusion occurring increases. With a target being invisible to the user, it is generally impossible to select, as the user will have no visual feedback as to the location of the cursor relative to the intended target.

As discussed in Section 2.3, existing techniques for overcoming such occlusion generally require explicit and separate steps on behalf of the user. This means that the actual selection cannot be performed in a single, fluid, interaction. This is a drawback which motivates us to find other strategies.

When a user wishes to select an occluded object, the occlusion introduces a problem of *access*, and not *discovery* [8]. It can, thus, be assumed that the user has a good conceptual model of the environment, knowing the general area of the intended target, so going through an explicit process of altering the viewing state of

the entire scene may be excessive. Under this assumption, we apply a design strategy of integrating visual enhancements into the selection technique itself. This allows occluded targets to be made visible, without the requirement of additional modes or auxiliary devices. The idea is to apply these enhancements only to targets in the vicinity of the selection cursor, almost as if the selection cursor were also controlling a magic lens [1]. We explain the specific details in the next section, when we describe the selection techniques which we design and evaluate in this paper.

4 SELECTION TECHNIQUES

In this section, we discuss the selection techniques which will be used in our evaluation, developed with the above mentioned design guidelines and strategies in mind. We applied these design strategies to both hand extension and ray casting metaphors, resulting in a 3D bubble cursor and an augmented depth ray.

4.1 3D Bubble Cursor

We first apply our design guidelines to the hand extension metaphor. If we simply increase the activation area of such a selection technique we are left with a volume cursor, where the user must capture the intended target inside the cursor volume to select it. However, we also require the activation area to be unambiguous, and with a volume cursor, multiple targets can fall within the cursor’s boundaries. To alleviate this problem, we implemented a 3D version of the bubble cursor, which dynamically resizes such that only the closest target falls within its boundaries. We render the bubble cursor as a grey semi-transparent sphere similar to the rendering of the silk cursor [30] (Figure 1a). When necessary, we render a second semi-transparent sphere around the captured target, such that it always appears to be fully contained by the cursor (Figure 1b). For added visual feedback, we highlight captured targets yellow. As with the 2D implementation, we also render a crosshair inside the bubble cursor to mark its center. The user controls the location of this crosshair by positioning the input device in 3D space. We refer the reader to the original bubble cursor work for a full description of the algorithm used to calculate the cursor radius [12].

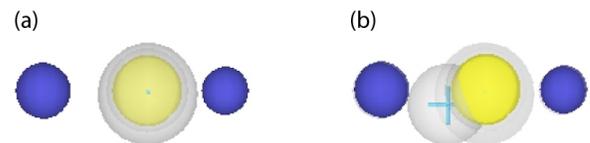


Figure 1. a) The 3D bubble cursor is rendered as a semi-transparent sphere which dynamically resizes such that it only captures the closest target, highlighted yellow. b) When necessary, a second sphere is rendered around the captured target so that it always appears to be completely contained by the bubble cursor.

Our hope is that this cursor will allow for efficient selections of targets in both sparse and dense environments. However, as per our design guidelines, we also wish for the technique to support the selection of targets which are occluded from the user’s viewpoint. To overcome such occlusions, we give the bubble cursor magic lens capabilities, such that targets in its vicinity become semi-transparent. To do so, we calculate the distance between the bubble cursor and each target, measured on the 2D image viewing plane. Any target within 4cm is rendered as semi-transparent, so that targets which may be initially occluded become visible (Figure 2a). This allows users to hone in on an occluded goal target as they approach it, assuming they know its general location (Figure 2b). To improve visual feedback, a solid opaque border is also displayed around the captured target.

It is important to note that this localized transparency function is only appropriate when users know the general region of their intended target. This is often the case when users are familiar with the scene that they are interacting with, and is an assumption we make in this paper. If the user needed to search for the target, global methods such as rotating the scene or switching to a different viewing mode would be more appropriate.

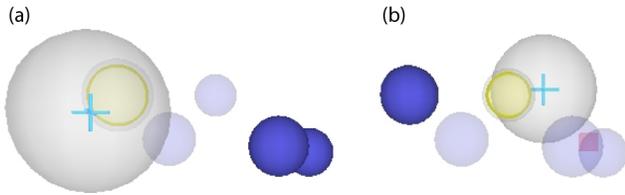


Figure 2. a) Targets in close proximity to the bubble cursor become semi-transparent. b) As the cursor approaches an occluded goal target (red cube) it becomes visible and can be selected.

4.2 Depth Ray

The 3D bubble cursor results from an application of our design strategies to the hand extension metaphor. Here, we apply these guidelines to the ray casting metaphor.

Without augmentation, ray cursors can already be thought of as having increased activation areas, since they can select any target along the length of the ray. A conic ray cursor further increases this activation area. With this increased activation area, multiple targets can simultaneously be captured. As such, we are, again, required to provide a disambiguation mechanism.

As previously discussed, the depth ray has been shown to provide an effective mechanism for disambiguating between multiple intersected targets [13]. The user controls a depth marker, which exists along the length of ray. Moving the hand forwards or backwards will make the depth marker move in the same manner. The object intersected by the ray cursor, which is closest to the depth marker, can be selected. We render the ray as a thin red cylinder, although a conic selection area, with an apex of 1°, originating from the user’s hand, is used for the selection test. As with the bubble cursor, the captured target is highlighted yellow and remaining targets intersected by the ray are highlighted green. Figure 3 shows a screenshot of our implementation.

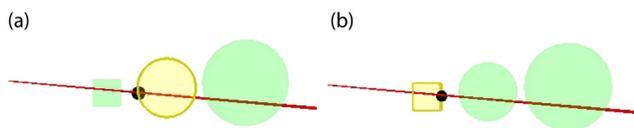


Figure 3. a) The depth ray selects the intersected target which is closest to the depth marker. b) The depth marker position can be controlled by moving the hand forwards or backwards.

To allow for the selection of occluded targets, we augment the depth ray with a similar transparency function used by the bubble cursor, using the distance between the targets and the ray, measured in the 3D environment.

4.3 Effective Widths

The effective width of a target can be defined as the size in motor space of a target’s activation boundaries. It has been shown that the effective width of a target plays a larger role than its visual boundaries on selection times [2, 12]. While the goal of this work is not to obtain a sound theoretical model of how each technique will perform, understanding the effective widths for each technique will help us form hypotheses on the relative performances of the two techniques.

The dynamic activation area of the bubble cursor divides the 3D environment into regions, such that there is exactly one target inside each region, with that target being closest to any point inside that region. These are also known as voronoi regions. The 3D bubble cursor increases the effective width of a target to its surrounding 3D voronoi region. In other words, to select a target, the center of the bubble cursor only needs to be positioned inside the target’s associated voronoi region (Figure 4a).

While the effective width of a target when using the bubble cursor is completely dependent on the location of the surrounding targets, this is not the case for the depth ray. For two of the dimensions, the effective width depends on the angular accuracy required to intersect the target. This depends on the target size, distance, and conic apex of the ray. The third dimension of effective width depends on the current trajectory of the ray, as the voronoi region of the goal target is based on only those targets currently intersected by the ray. This voronoi region, projected onto the length of the ray, defines the segment of the ray where the depth marker can select the target. It is the length of this segment which can be thought of as the effective width. In some situations, this will result in a similar effective width to when using the bubble cursor (Figure 4b). However, if the ray is positioned to avoid certain surrounding targets, the effective width can be larger (Figure 4c). This could potentially make the depth ray a faster technique than the bubble cursor.

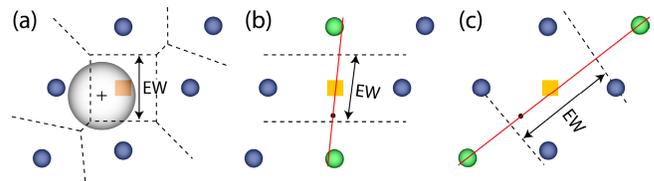


Figure 4. a) The bubble cursor divides the environment into 3D voronoi regions. The effective width of each target is defined by its corresponding voronoi region. b) When using the depth ray, the voronoi region is based only on the intersected targets. c) Changing the position of the ray can change a target’s effective width.

5 EXPERIMENT

We have applied our design strategies to both hand extension and ray casting metaphors, giving us the 3D bubble cursor and depth ray techniques, both augmented to allow for the selection of occluded targets. Our hope is that these techniques will allow for efficient selections in both sparse and dense target environments, and for both visible and occluded targets. Previous work tells us that the depth ray can be efficient in dense environments [13], but it is unknown how exactly the density will affect performance. A better understanding exists for the effect of density on the 2D form of the bubble cursor [12], but no such understanding exists for its 3D counterpart which we have introduced in this paper.

In this section, we present a formal experiment to evaluate these two techniques, where we manipulate both the environment density and goal target visibility in a 3D VR environment. One goal of the experiment is, thus, to gain an understanding of how these variables affect selections. However, another goal is to compare the relative performance of these two techniques, and also provide a comparison to a baseline technique.

In some of the experimental environments which we will use, it would be impossible to select a target with a naive implementation of the ray cursor, because of occlusions. For that reason, we use a 3D point cursor for a baseline comparison technique. The point cursor is rendered as a 3D crosshair, the same as the crosshair drawn in the center of the 3D bubble cursor, described in Section 4.1. In order to select a target, the center of

the crosshair must be positioned inside of it. To allow for the selection of occluded targets, we augment the point cursor with the same transparency functionality used for the other techniques. A true baseline technique would not have this built-in functionality, and would require users to either rotate the scene or switch viewing modes to select an occluded target. However, including the transparency function minimizes hardware requirements, and provides for the fairest possible comparison.

We hypothesize that an increased environment density will slow selection times for the depth ray and bubble cursor, as an increased density will decrease effective target widths. However, the density should not have as much effect on a standard point cursor, as it has no effect on the motor space size of the target's activation boundaries. We also hypothesize that even with the transparency functions which our techniques use, it will still take longer to select targets which are occluded, as the visual feedback is initially reduced. However, our hope is that our techniques will reduce the overhead cost when the targets are not initially visible. Lastly, in comparing the three techniques, we hypothesize that the depth ray and bubble cursor will outperform the point cursor. The relative performance of the bubble cursor and depth ray is harder to predict, and will be an interesting result to analyze.

5.1 Apparatus

The display we used was a 21-inch Hitachi CM813ETPlus monitor combined with StereoGraphics CrystalEyes 3D LCD shutter glasses for stereo viewing. Left and right eye images were provided using quad buffering at a refresh rate of 60Hz per eye which was coordinated through the shutter glasses by an infrared transmitter positioned on top of the monitor.

For input a Polhemus Fastrak 6 DOF magnetic tracker was used, encapsulated in a handheld device. The device was equipped with a single button. The tracker was updated at 120Hz with a precision of less than one millimetre. The input device controlled each of the three cursors with an absolute 1 to 1 mapping. Participants stood during the course of the experiment, and the display was raised to be roughly shoulder level. Figure 5 illustrates the experiment apparatus.

5.2 Participants

Eleven male and one female unpaid volunteers, ranging in age from 20 to 28, served as participants in this experiment. Participants were screened using a stereopsis test in which they had to order objects according to their depth. All participants were right handed and used their right hand to control the input device.

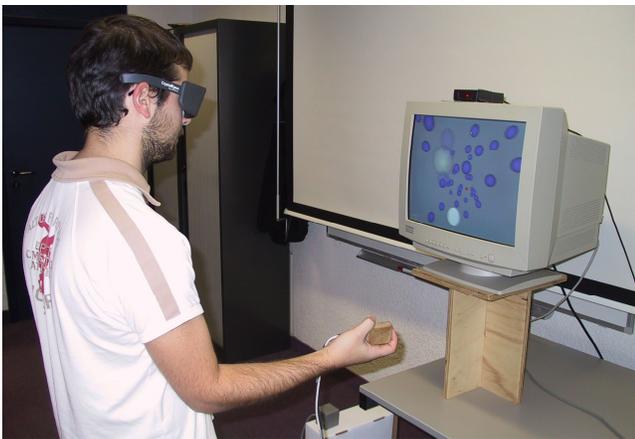


Figure 5. The experiment apparatus.

5.3 Procedure

A 3D static target acquisition task was used for the experiments. The scene consisted of a start target, goal target, and 45 distracter targets. The start target was rendered as a white sphere, the goal target as a red cube, and the distracter targets as blue spheres. To complete the task, participants first selected the start target, centered at the front of the display space. Once selected, this target would disappear, and the participant would then have to select the goal target. The position of the start target was static, and the goal target was positioned in a random location, such that the distance between the start and goal targets was 20cm in 3D space. The positions of the distracter targets were random, with the constraint that they did not intersect. The radius for the distracter targets were randomly assigned values between 0.75cm and 1.5cm. Figure 6 illustrates the experiment environment.

Each trial started with a fade-in of the scene. Initially, only the goal target would be visible. After 500ms the distracter targets would fade in over a duration of 2 seconds. This would give users an understanding of the general location of the goal target for the occluded conditions. Once all targets appeared, the user could select the start target to begin the trial. Users were instructed to select the goal target as fast as possible while minimizing their errors. Users had to successfully select the goal target before the trial could end.

We included the target highlighting discussed for our selection techniques. Captured targets were rendered yellow with a solid opaque border. Additionally, other targets intersected by the depth ray were highlighted green.

5.4 Independent Variables

The main factors we study in this experiment are the selection technique, environment density, target visibility, and target size.

For environment density, we are mostly concerned with the immediate surroundings of the goal target, as this will affect the effective width for the bubble cursor and depth ray. To control this variable, we carefully position six distracter targets around the goal target. Two distracter targets are placed along the direction of movement, defined as the vector between the start and goal targets, one before and one after the goal target. Perpendicular to this vector, targets are then placed above, below, and to either side of the goal target, forming a cube shaped voronoi region. We can control the size of the resulting voronoi region by changing the distance between these six distracter targets and the goal target, measured from their closest edge. We call this distance variable density spacing, DS (Figure 7).

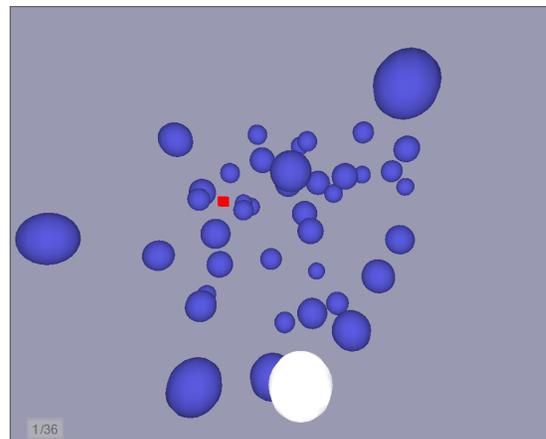


Figure 6. The experiment environment.

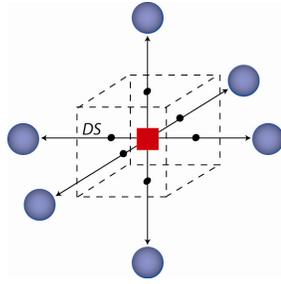


Figure 7. Six distracter targets are carefully positioned around the goal target, creating a cube-shaped voronoi region. The distance between these targets and goal target is the density spacing (DS).

For goal target visibility, we tested fully visible and fully occluded conditions. In both conditions, we ensured that none of the six surrounding targets were occluding the goal target. To ensure this, in some cases, we rotated the entire voronoi region. For the visible condition, we ensured that no other distracter targets occluded the goal target. For the occluded condition, we ensured that two distracter targets partially occluded the goal target, such that together the goal target was completely occluded.

5.5 Design

A repeated measures within-participant design was used. The independent variables were the cursor type CT (point cursor, bubble cursor, and depth ray), density spacing DS (1cm, 2.5cm, 5cm), visibility condition VC (visible, occluded), and target size $SIZE$ (0.75cm, 1.5cm). A fully crossed design resulted in 36 combinations of CT , DS , VC , and $SIZE$.

Each participant performed the experiment in one session lasting about 70 minutes. The session was broken up by the three cursor types, with 3 blocks appearing for each of the techniques. Within each block, the 12 combinations of DS , VC and $SIZE$ were repeated 3 times in a random order, for a total of 36 trials. Prior to the experiment, 36 environments were randomly generated for these trials, and were the same across all participants. The cursor ordering was fully counterbalanced across the 12 participants, with 2 participants randomly assigned to each of the 6 unique orderings. Before each cursor type, participants were given several warm-up trials to familiarize themselves with the selection technique.

5.6 Results

5.6.1 Trial Completion Time

In our analysis of trial completion time, we discarded trials in which errors occurred, and removed outliers that were more than 3 standard deviations from the group mean (1.7% of the data).

Repeated measures of analysis showed main effects for CT ($F_{2, 22} = 383$), $SIZE$ ($F_{1, 11} = 277$), VC ($F_{1, 11} = 330$), and DS ($F_{2, 22} = 68.7$), all at the $p < .0001$ level. Average trial completion times were 4.59s for the point cursor, 3.10s for the bubble cursor, and 2.85s for the depth ray. Post hoc comparisons showed that both the depth ray and bubble cursor were significantly faster than the point cursor ($p < .0001$), and that the depth ray was significantly faster than the bubble cursor ($p < .005$).

We also found that CT had significant interaction effects with $SIZE$ ($F_{2, 22} = 12.8$, $p < .0001$), VC ($F_{2, 22} = 118$, $p < .0001$), and DS ($F_{4, 44} = 3.97$, $p < .005$), showing that each of these independent variables affected the three techniques differently. We now provide a discussion on each of these observed interaction effects.

The interaction between CT and DS is illustrated in Figure 8. The general trend, as expected, is that the selection times are reduced with increased density spacing. There are, however, two interesting results which are illustrated.

First, the density had a significant effect on movement times for the point cursor ($p < .0001$). This is somewhat surprising since it should only be the target size which constrains a selection with the point cursor. A possible explanation is inadequate visual feedback. In the dense conditions, it may have been more difficult to discern when the goal target had been highlighted.

Second, it is interesting to note that for the depth ray, increasing the density spacing from 2.5 to 5cm did not significantly reduce movement times. In fact, the movement times actually increased, although the effect was not significant. In contrast, the bubble cursor completion times decreased significantly for each increase in density spacing ($p < .05$). While the data for the depth ray may at first seem counterintuitive, we must recall that the effective width of the goal target when using the depth ray is not completely determined by the density spacing value. It also depends on the angle of approach that the ray takes on (Figure 4), so it may be the case that users were not taking on optimal approach angles for the condition $DS = 5$ cm. This could have been for many reasons, one of which being that the randomly generated environments were different for each density value.

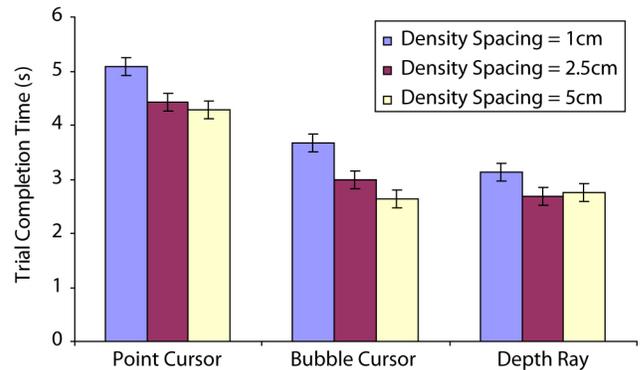


Figure 8. Technique completion times by density.

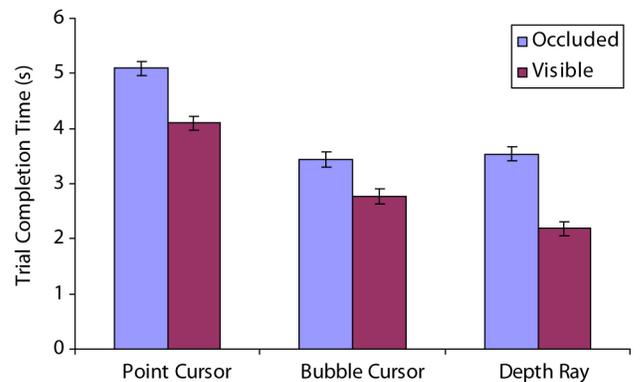


Figure 9. Technique completion times by visibility condition.

Figure 9 illustrates the interaction between CT and VC . As can be seen, the occluded condition increases trial completion times for all three cursors ($p < .0001$), with an average increase in completion times of 1.01s. From this figure, we can also see that the depth ray was the fastest technique overall because of its superiority in the visible condition. Post hoc multiple means comparison shows that in this condition, the depth ray is

significantly faster than the bubble cursor ($p < .0001$), whereas in the occluded condition, there is no difference between the two techniques. It may be the case that the overhead cost introduced by the occluded condition outweighs the differences between the two techniques, so their differences are not apparent.

Finally, the interaction effect between *CT* and *SIZE* is illustrated in Figure 10. As can be seen, the size had its most dramatic impact on the point cursor, which is the only technique constrained by the dimensions of the actual target. However, the size also had a significant effect on the bubble cursor ($p < .0001$), with completion times of 3.4s for *SIZE* = 0.75cm and 2.8s for *SIZE* = 1.5cm. This is somewhat surprising, since selections with this technique are constrained by the proximity of the surrounding targets, and not the actual target size. This indicates that users were being drawn towards the visible borders of the targets, not taking full advantage of the target's effective width. This result is consistent with observations of the 2D bubble cursor, which showed a similar result, although not as strong. The effect may be increased in our 3D task because the visual feedback is not as clear cut, as multiple objects could be layered of the cursor.

In the 2D case it was shown that the size of the target will have more effect on selection times when the surrounding targets are closer. To determine if this was the cause for the effect we were observing, we looked at the interaction between *SIZE* and *DS* for the bubble cursor condition. Indeed, the interaction effect was significant ($F_{2, 22} = 17, p < .0001$), illustrated in Figure 11. It can be seen that the effect of size increased when surrounding targets were closer to the goal target. Post hoc comparisons show that size had a significant effect for *DS* = 1cm and *DS* = 2.5cm ($p < .0001$), but not for *DS* = 5cm.

As for the depth ray, target size had no significant effect, indicating the visual feedback of when targets were captured may have been better than for the bubble cursor, reducing unnecessary movements towards the boundaries of the goal target.

5.6.2 Learning

A learning effect was observed in our experiment, with the block number significantly affecting trial completion times ($F_{2, 22} = 23.8, p < .0001$). Performance improved over time, with average completion times of 3.79s, 3.43s, and 3.33s for blocks 1, 2, and 3 respectively. Post hoc comparisons shows that Block 1 was significantly slower than Blocks 2 and 3 ($p < .0001$), but Blocks 2 and 3 were not significantly different.

There was also a significant interaction between the block number and *CT* ($F_{4, 44} = 2.54, p < .05$). The effects, illustrated in Figure 12, indicate that there is little overhead cost involved with learning the techniques. It can be seen that the most learning occurs with the point cursor. It is also interesting to note that the bubble cursor and depth ray are almost equivalent in the first block, and differ significant only in the last block ($p < .05$). This may indicate that users learned to take optimal approaches to the goal target using the depth ray, to maximize the effective width.

5.6.3 Input Device Footprint

The input device footprint is defined as the length of the total path which the input device traveled to complete each trial. The cursor type had a significant effect on the input device footprint ($F_{2, 22} = 210, p < .0001$). The footprints were 31.2cm for the point cursor, 26.9cm for the bubble cursor, and 22.4cm for the depth ray, all significantly different ($p < .0001$). The depth ray likely had the lowest value because, for this technique, users are not required to move their hand towards a specific 3D location.

5.6.4 Error Rates

For this task, errors were defined as trials in which users selected the wrong target before selecting the goal target. The overall error rate for the experiment was 2%. For the point cursor no errors were reported. This shows that users could use the visual feedback provided to tell when the goal target was highlighted. The bubble cursor had an error rate of 2.2% and the depth ray 3.4%, for which there was no statistical difference.

5.6.5 Subjective Feedback

Overall, participants found the bubble cursor and depth ray easy to use, preferring these techniques over the point cursor. Participants who preferred the depth ray noted that less movement was required, and that they could aim from any angle. All 12 participants responded that they liked the transparency function.

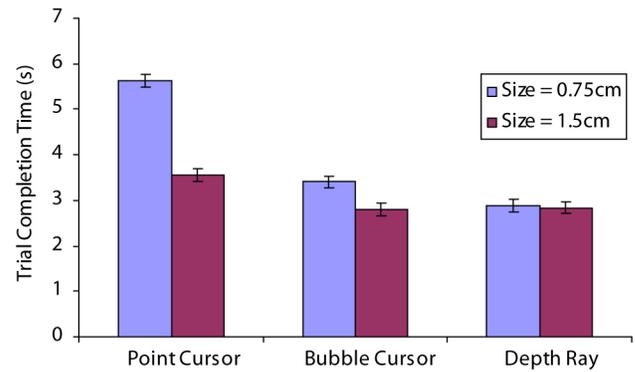


Figure 10. Technique completion times by target size.

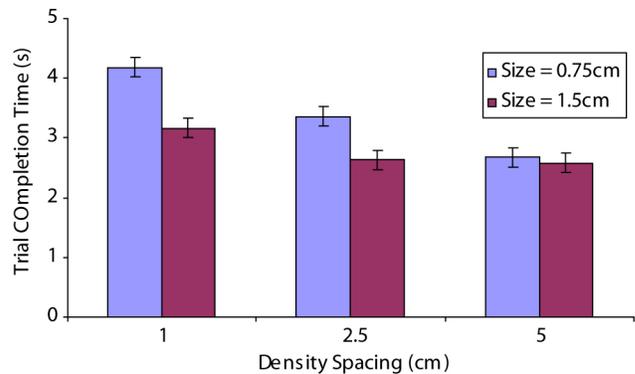


Figure 11. Bubble cursor times by size and density spacing.

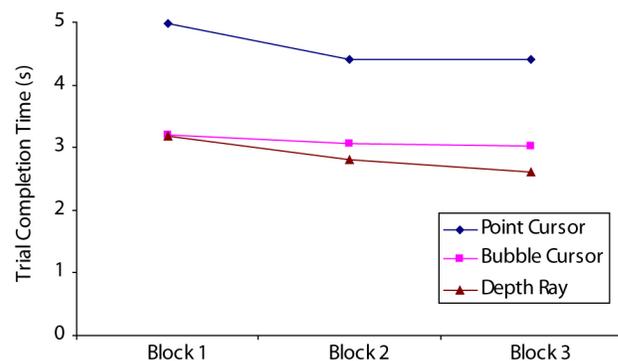


Figure 12. Learning effects for each of the techniques.

6 DISCUSSION

The results of our studies have some important implications to user interfaces for VR environments. We have discussed the design of two beneficial techniques for object selection. While the depth ray was an existing technique for volumetric displays, we have provided its first VR implementation. Similarly we have provided the first implementation of a 3D bubble cursor. Furthermore, these techniques are both original in that they are augmented to allow for the selection of occluded targets.

The depth ray which we presented was the most successful of the techniques evaluated. Our discussion of effective widths may explain why the depth ray outperformed the bubble cursor. For the depth ray, the effective width of a target could be increased by changing the approach angle of the ray. Another benefit of the depth ray was that it lowered the input device footprint, which could minimize arm fatigue over extended usage. The poor performance of the 3D point cursor further validates our design of new techniques, as well as our discussed design guidelines.

Our results also indicated that with improved visual feedback, the bubble cursor may have performed just as well as the depth ray. It may also be interesting to explore alternate forms of feedback, including possible multimodal strategies [26].

It is not surprising that our results showed that for all techniques, occluded targets took longer to select. However, it is important to note that our transparency function did enable users to minimize the overhead cost for such tasks to just one second. If a more traditional approach for making the target visible were used, such as using a hotkey to switch viewing modes or rotating the scene, we would expect to see a similar, if not greater, overhead cost, with the added drawback of extra buttons or input devices being required. This transparency function could also be used for higher level tasks such as the exploration of dense and complex environments.

7 CONCLUSION

We have presented an in-depth evaluation of two new selection techniques which support selections in dense target environments, and of targets which are fully obscured from the user's viewpoint. Both of these techniques outperformed a baseline point cursor in our experimental task, providing faster selection times and lower device footprints. In summary, we believe that the contributions of this work will be valuable for future designers of interactive 3D applications.

8 ACKNOWLEDGEMENTS

We thank the experiment participants and Tom De Weyer for his assistance in our implementation. Part of the research at EDM is funded by ERDF (European Regional Development Fund), the Flemish Government and the Flemish Interdisciplinary institute for Broadband technology (IBBT).

REFERENCES

- [1] Bier, E., M. Stone, K. Pier, W. Buxton and T. DeRose (1993). Toolglass and Magic Lenses: The see-through interface. *ACM SIGGRAPH*. p. 73-80.
- [2] Blanch, R., Y. Guiard and M. Beaudouin-Lafon (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *ACM CHI*. p. 519-525.
- [3] Bowman, D. A., D. B. Johnson and L. F. Hodges (1999). Testbed evaluation of virtual environment interaction. *ACM VRST*. p. 26-33.
- [4] Bowman, D. A., E. Kruijff, J. J. LaViola and I. Poupyrev (2004). *3D User Interfaces: Theory and Practice*, Addison Wesley Longman Publishing Co., Inc.
- [5] Buxton, W. and G. W. Fitzmaurice (1998). HMD's, Caves, & Chameleon: A human-centric analysis of interaction in virtual space. *Computer Graphics, The SIGGRAPH Quarterly* 32(4): 64-68.
- [6] Carpendale, S., D. Cowperthwaite and D. Fracchia (1997). Extending distortion viewing from 2D to 3D. *IEEE Computer Graphics and Applications*. 17(4): 42-51.
- [7] Elmqvist, N. (2005). BalloonProbe: reducing occlusion in 3D using interactive space distortion. *ACM VRST*. p. 134-137.
- [8] Elmqvist, N. and P. Tsigas (2006). View projection animation for occlusion reduction. *ACM AVI*. p. 471-475.
- [9] Forsberg, A., K. Herndon and R. Zeleznik (1996). Aperture based selection for immersive virtual environments. *ACM UIST*. p. 95-96.
- [10] Grossman, T. and R. Balakrishnan (2004). Pointing at trivariate targets in 3D environments. *ACM CHI*. p. 447-454.
- [11] Grossman, T., D. Wigdor and R. Balakrishnan (2004). Multi finger gestural interaction with 3D volumetric displays. *ACM UIST*. p. 61-70.
- [12] Grossman, T. and R. Balakrishnan (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *ACM CHI*. p. 281-290.
- [13] Grossman, T. and R. Balakrishnan (2006). The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. *ACM UIST*. p. 3-12.
- [14] Hinckley, K., R. Pausch, J. C. Goble and N. Kassell (1994). A survey of design issues in spatial input. *ACM UIST*. p. 213-222.
- [15] Liang, J. and M. Green (1994). JDCAD: A highly interactive 3D modeling system. *Computers and Graphics* 18(4): 499-506.
- [16] Livingston, M. A. et al. (2003). Resolving Multiple Occluded Layers in Augmented Reality. *Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*. p. 56.
- [17] McGuffin, M., L. Tancau and R. Balakrishnan (2003). Using deformations for browsing volumetric data. *IEEE Visualization*. p. 401-408.
- [18] Mine, M. (1995). Virtual environment interaction techniques, UNC Chapel Hill Computer Science Technical Report TR95-018.
- [19] Olwal, A. and S. Feiner (2003). The Flexible Pointer - An Interaction Technique for Selection in Augmented and Virtual Reality. *Conference Supplement of ACM UIST*. p. 81-82.
- [20] Pierce, J., A. Forsberg, M. Conway, S. Hong and R. Zeleznik (1997). Image plane interaction techniques in 3D immersive environments. *ACM Symposium on Interactive 3D Graphics*. p. 39-43.
- [21] Poupyrev, I., M. Billinghurst, S. Weghorst and T. Ichikawa (1996). The go-go interaction technique: non-linear mapping for direct manipulation in VR. *ACM UIST*. p. 79-80.
- [22] Poupyrev, I., S. Weghorst, M. Billinghurst and T. Ichikawa (1998). Egocentric Object Manipulation in Virtual Environments: Evaluation of Interaction Techniques. *Computer Graphics Forum* 17(3): 41-52.
- [23] Ramos, G., et al. (2006). Tumble! Splat! helping users access and manipulate occluded content in 2D drawings. *ACM AVI*. p. 428-435.
- [24] Steed, A. and C. Parker (2004). 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays. *International Immersive Projection Technology Workshop*.
- [25] Steed, A. (2006). Towards a General Model for Selection in Virtual Environments. *IEEE 3DUI*. p. 103-110.
- [26] Vanacken, L., C. Raymaekers and K. Coninx (2006). Evaluating the Influence of Multimodal Feedback on Egocentric Selection Metaphors in Virtual Environments. *First International Workshop on Haptic Audio Interaction Design*. 12 - 23.
- [27] Ware, C., K. Arthur and K. S. Booth (1993). Fish tank virtual reality. *ACM CHI*. p. 37-42.
- [28] Ware, C. and G. Franck (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph.* 15(2): 121-140.
- [29] Wyss, H. P., R. Blach and M. Bues (2006). iSith - Intersection-based Spatial Interaction for Two Hands. *IEEE 3DUI*. p. 59-61.
- [30] Zhai, S., W. Buxton and P. Milgram (1994). The "Silk Cursor": Investigating transparency for 3D target acquisition. *ACM CHI*. p. 459-464.