Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality

Yiqin Lu * Chun Yu † Yuanchun Shi ‡

Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education, China

2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) 978-1-7281-5608-8/20/$31.00 ©2020 IEEE 10.1109/VR46266.2020.1581165829725

ABSTRACT

Ray-casting, i.e., a ray cast from a hand-held controller to select targets, is widely used in 3D environments. Inspired by the bubble cursor [12] which dynamically resizes its selection range on 2D surfaces, we investigate a bubble mechanism for ray-casting in virtual reality. Bubble mechanism identifies the target nearest to the ray, with which users do not have to accurately shoot through the target. We first design the criterion of selection and the visual feedback of the bubble. We then conduct two experiments to evaluate ray-casting techniques with bubble mechanism in both simple and complicated 3D target acquisition tasks. Results show the bubble mechanism significantly improves ray-casting on both performance and preference, and our Bubble Ray technique with angular distance definition is competitive compared with other target acquisition techniques. We also discuss potential improvements to show more practical implementations of ray-casting with bubble mechanism.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Human-centered computing—Interaction design—Interaction design process and methods—User interface design

1 INTRODUCTION

Target acquisition is one of the most elementary interactions in 3D environments. Providing fast and accurate target acquisition is important in designing virtual reality games and tools. Since a 3D virtual environment can be as large as the user's field of view, ray-casting [16, 21], i.e., a ray cast from the ray source to the infinity, has been widely used in virtual reality. Ray-casting has been applied in many commercial virtual reality devices (e.g., HTC Vive), with which the user holds a position-tracked controller to select by making the ray go through the target (Figure 1a). However, ray-casting will be unstable when selecting small and distant targets due to the unintentional tremor from the user's hand. This problem leads to many researches to improve the selection performance of ray-casting.

Bubble mechanism is first proposed from the bubble cursor [12] for 2D target acquisition. The bubble cursor dynamically resizes its selection range (bubble) of the cursor and guarantees only the nearest target is contained by the bubble (Figure 1b). Bubble mechanism allows selecting the nearest target without directly hitting the target, which utilizes the empty space around targets and improves the selection performance. Bubble mechanism has been shown promising in 2D graphical interfaces and been extended into the 3D bubble cursor [38], however, it has not been explored for ray-casting.

In this work, we investigate how the bubble mechanism can improve ray-casting for 3D target acquisition in virtual reality. We first define two distance definitions of the bubble mechanism - Euclidean and Angular, to measure the relationship between the ray and the target. We then design a disc-shaped bubble tangent to the target as the visual feedback using an iterative design process. Next, we evaluate ray-casting techniques with bubble mechanism by comparing them with other 3D target acquisition techniques in both simple and complicated tasks. Results consistently show the technique with angular distance definition has high performance in general cases with less selection time, lower error rate and better user experience, and it is also able to provide relatively stable selection when meeting dense or occlusive cases. At last, we discuss potential improvements according to the limitation of the bubble mechanism. Our findings of the bubble mechanism for ray-casting make an incremental contribution to the area of 3D target acquisition and benefit the community of virtual reality.

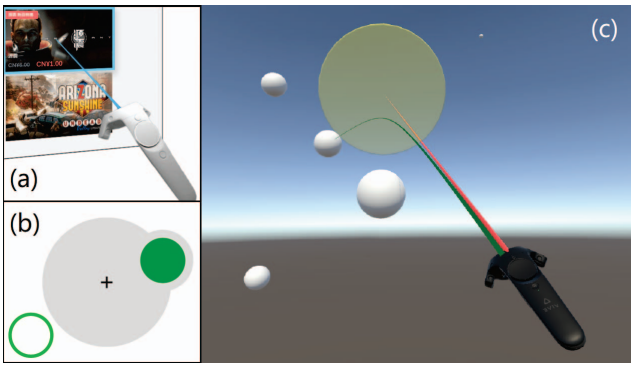

Figure 1: (a) Ray-casting. (b) Bubble Cursor [12]. (c) Ray-casting with bubble mechanism. The ray (in red) is cast from the controller. The green curve is the target indicator directing to the selected target. Yellow disc is the bubble from the user's view, which is tangent to the selected target.

2 RELATED WORK

Related works will be introduced in two subsections. In the first subsection, we will list existing ray-casting techniques for 3D target acquisition and emphasize their features. In the second subsection, we will introduce the bubble mechanism and show its variants and applications in 2D and 3D cases. To our knowledge, no work has been done to investigate the bubble mechanism for ray-casting on 3D target acquisition tasks in literature.

2.1 Ray-Casting Target Acquisition Technique

Ray-casting metaphor, often mentioned as "pointing" in some non-virtual cases, is that a ray cast from the ray source to select targets intersected with the ray. Ray-casting with a hand-held ray emitter (5 DOF) was first proposed as a "laser gun" or "laser pointer" [16, 21]. Due to the intuitiveness and user-friendliness, hand-held ray-casting has been widely used in virtual reality researches and platforms for a

*e-mail: lu-yq16@mails.tsinghua.edu.cn
†e-mail: chunyu@tsinghua.edu.cn, the corresponding author
‡e-mail: shiyc@tsinghua.edu.cn

2642-5254/20/$31.00 ©2020 IEEE
DOI 10.1109/VR46266.2020.00-83

long time, including many commercial VR devices (e.g., HTC Vive controller). Another implementation of ray-casting is to fix the ray source on the head while the head or gaze rotation is mapped to the orientation of the ray (2 DOF). Previous works have investigated gaze ray [24], ray-casting through an aperture circle [11] and seeing through hand poses [29].

The user suffers from the hand tremor problem [26] when using ray-casting techniques to select small and distant targets. To improve the stability of the selection, "spotlight", i.e., cone-like ray [21] was proposed for a larger range of the selection. However, the enlarged range of the selection may involve multiple targets and bring target ambiguity problems [13,32]. Some heuristic techniques [7,13,27,33] employed a prediction algorithm with spatial and temporal models to determine the most possible target. The "heuristics" is always implicit to the user and lacks visual feedback, hence it is not natural for the user to infer the current status of the selection. Snap-to strategy [11] was mentioned as the "sticky" feature which allows the ray to stick to the target when the user's hand is trembling, but the performance was not validated.

When a ray goes through multiple targets, it also causes target ambiguity problems [13, 32] that the user has to specify the desired target. This problem will be more serious when targets are occluded from the field of view or when using cone-like ray-casting. "Shadow Cone" [34] allowed the user to move the hand in different angles to remove the undesired targets. "Depth Ray" [13] attached an additional cursor along the ray to enable unique target selection through the depth control. "Go-Go" [30] and its advanced version [2] allowed the user to stretch his/her arm to grab targets in the direction of the ray using a nonlinear mapping between the real and the virtual hands. Two-step techniques have been proposed in many previous works, which allow the user to specify the unique target from intersected candidates after the ray selects multiple targets. The form of specifying is usually a pop-up menu for the secondary disambiguation like floating menu [6], "Flower Ray" [13], "SQUAD" [18] or the menu selection on a touchscreen [8].

Many studies [3,22,31,38] have compared ray-casting techniques with other 3D target acquisition techniques. Results showed 3D target acquisition was complicated so that the performance of techniques varied in different experimental settings. Ray-casting techniques have competitive performances in selecting remote targets, but they become awkward when targets are very close and touchable. Some previous works [4, 32] have evaluated environment density or target visibility when using ray-casting techniques. Results showed the original ray-casting technique was fast but error-prone and needed to be improved to tackle extreme cases. We believe the bubble mechanism is worth exploring to improve ray-casting in that case.

2.2 Bubble Mechanism

The bubble cursor [12] is a promising target acquisition technique in 2D graphical interfaces. It dynamically resizes its activation area of selecting and guarantees that exactly one target will fall within its bubble at any time. The bubble cursor improves the flexibility upon the area cursor [42] to provide an adaptive "bubble mechanism", which efficiently utilizes the proximity of surrounding targets to reduce the movement of the cursor according to 2D Fitts' law [10,23]. Previous works have proposed some incremental changes upon the bubble cursor, such as magnifying the nearby region when targets are small and dense [25] and controlling the bubble effect by the speed of the cursor [5].

The bubble mechanism is essentially based on the Voronoi partition of the control and display space. Extending 2D areal bubble cursor to 3D volumetric bubble cursor [38] has been already proposed. The implementation of a 3D bubble cursor is typically based on a position-tracked device held by the user's hand (3 DOF). The result showed 3D bubble cursor had a competitive performance compared with the ray-casting technique. However, the user's weak perception of the depth in virtual reality [20] and the tracking capability of devices limit the use of a 3D bubble cursor in a small range in front of the human body.

Previous works have also investigated using ray-casting to control the bubble cursor on 2D remote surfaces. A comparison between Voronoi-based target acquisition techniques [14] showed the target expansion technique had a little better performance than the remote bubble cursor. Semantic pointing [9] simplified the 3D environment from the user's view into a 2D desktop and enabled a 2D bubble cursor to select targets. Speech-filtered bubble ray [37] showed the feasibility of using speech commands to specify the target in remote pointing tasks. However, no work has been found to explore the bubble mechanism for ray-casting in 3D virtual environments with targets in various depths.

3 Designing Bubble Mechanism for Ray-casting

In this section, we will design ray-casting techniques in virtual reality augmented by the bubble mechanism. Two issues will be addressed. The first issue is the criterion of target specification, that is, how to determine which target should be selected when the ray does not go through any target. The second issue is how to design the visual feedback of the bubble to provide a better interactive experience.

3.1 Criterion of Target Specification

In the design of the 2D bubble cursor [12], the algorithm chooses the target with the minimum Euclidean distance between the cursor and its boundary. This mechanism ensures there is exactly one target selected at any time. The 3D bubble cursor [38] directly defines the 3D Euclidean distance as the criterion. However, how to define "the target with the minimum distance" for ray-casting (2 DOF or 5 DOF) is not as easy as the cursor (3 DOF). Similar to the previous work [35], we come up with two definitions of the distance:

Euclidean: is defined as the minimum Euclidean distance between the ray and the target boundary (Figure 2a). This definition is derived from the original idea of the bubble cursor, straightforward demonstrating the distance in 3D spaces.

Angular: is defined as the angle between the ray and the target boundary, centered on the user's hand (Figure 2b). More specifically, we project the target and the ray on a plane determined by the ray vector and the target center, and we define the 2D angle on the plane as the angular distance. The angular definition has been discussed in the previous work [19] and showed contributing to 3D pointing tasks.

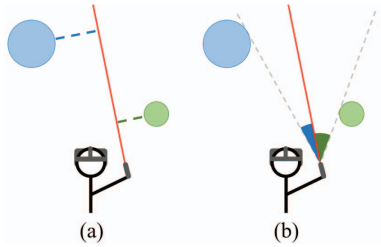

Figure 2: (a) Euclidean: blue or green dash line indicates the Euclidean distance of a target. (b) Angular: blue or green angle within gray dash lines indicates the angular distance.

Euclidean distance directly represents the distance in 3D spaces, but distance perception in the immersive environment of virtual reality is limited due to poor depth perception. The user may need to be very precise to succeed in selecting a distant target when there is a near distractive target. Angular distance better measures the projected distance from the user's field of view, but there is parallax between the user's eyes and hand. We consider angular distance is

the better criterion for ray-casting with the bubble mechanism, but we will evaluate techniques with both these distance definitions in the next section to provide empirical evidence.

Note the distance definition only works when the ray does not go through any target. If the ray goes through multiple targets, we consider the first intersected target is selected.

3.2 Visual Design of Bubble

The visual presentation of the bubble is an important real-time feedback for the user to estimate the status of the selection and infer the next movement. Good visual guidance may lead to better user performance and experience [15]. We follow an iterative design process to refine the visual feedback of the bubble mechanism. That is, we invite users to try our implementations of the technique with the bubble feedback and collect their comments for redesign and improvement, iteratively.

3.2.1 Bubble Size

The 3D bubble cursor [38] has introduced the way of rendering two semi-transparent spheres to show the bubbles, while the benefit of semi-transparent rendering was first shown in "Silk Cursor" [38, 43]. In previous works (both 2D plane bubbles and 3D semi-transparent spheres), the radius of the bubble was determined by Euclidean distances of the first and second nearest targets. The size of the bubble was set to just contain the nearest target but separate from the others. Unlike the bubble cursor where the bubble is centered on the cursor, the bubble for ray-casting has not been defined in previous works.

The most straightforward idea is to design a spherical semi-transparent bubble centered on the ray which contains the nearest target. However, the 3D environment has an additional depth axis, thus, targets behind the bubble from the user's view will be occluded by the bubble. Although the design of semi-transparency provides a see-through feature that allows the user to see the targets behind, the visual cues (e.g., shadows or textures), color differentiation and depth perception may still be affected [36].

Therefore, the size of the bubble should be as small as possible to reduce the occlusion. At this point, we defined the radius of the bubble to be the minimum distance between the ray and the boundary of the target, while the center of the bubble was placed at the best position along the ray corresponding to the minimum distance (Figure 3a). From the user's view, the bubble is tangent to the nearest target.

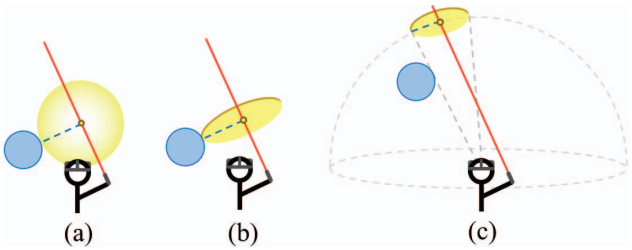

Figure 3: Different designs of the bubble. (a) Spherical bubble. (b) Disc-shaped bubble. (c) The bubble rendered on a distant spherical surface centered on the user's eyes, which is exactly tangent to the target from user's view.

3.2.2 Bubble Shape

Users from the pilot trial found the 3D spherical bubble might have a huge volume that could not be accepted in virtual reality. As the distance between the nearest target and the ray increased, the bubble was going to expand to a very large scale, as our invited user said, *"The bubble becomes so big that I even stay inside it"*. Also, the bubble might contain small and neighbored targets when the nearest target was determined in angular distance.

Considering the visual disturbance caused by the volume of the bubble, we choose to render only a 2D disc-shaped bubble which is tangent to the nearest target (Figure 3b). The disc-shaped bubble eliminates the redundancy on the depth axis but provides similar angular width as the spherical bubble from the user's view [19].

3.2.3 Transition

In the following user trial, most users preferred our tangent disc-shaped design compared with the spherical one. But some of them responded, *"Sometimes it is annoying when the bubble suddenly jumps between distinct depths when the selected target is changed"*. One user said, *"The bubble should be improved so that I don't have to look for the new location of the bubble"*. Another user said, *"The transition of the bubble from one target to another is not as smooth as that in a 2D bubble cursor"*.

To avoid annoying jumping, we design a new way to render the bubble. Assuming that the user's field of view is regarded as a large spherical surface centered on the user's eyes, we render the bubble on that spherical surface with a specific radius to make the bubble exactly tangent to the target from the user's view (Figure 3c). This rendering method is similar to projecting all targets on the large spherical surface and executing a 2D bubble cursor. When the selected target changes, the bubble will smoothly attach to the new target and keep the tangency.

3.2.4 Target Indication

Users from the next trial were satisfied with our new transition design. One of them suggested, *"When the bubble is almost tangent to two targets, I'm confused if I select the right one. It will be better to point out which target I have selected"*.

We augment the design with a target indicator in the form of a curve ray connected between the ray source and the selected target. This concept was presented in many previous works [6, 7, 35]. Generally, we use a Bezier curve to render it. The final design is shown in Figure 1c.

4 Experiment 1: Performance Evaluation of Ray-casting with Bubble Mechanism

In this experiment, we will evaluate the performance of ray-casting techniques with the bubble mechanism design in 3D target acquisition tasks. The goal of this experiment is to investigate how the bubble mechanism improves the speed and accuracy of ray-casting and whether it is competitive compared with other target acquisition techniques.

4.1 Participants and Apparatus

We recruited twelve participants (one female and eleven males, aged 19-25) from the university campus. All participants were right-handed. Seven of them had experience with ray-casting techniques in virtual reality before.

This experimental platform was running on a 4.00GHz Intel Core-i7 PC running Windows 10. The experimental scenes were built in Unity 5.6.0. We used HTC Vive VR headset as the device to render the virtual environment for users. We also used a position-tracked Vive controller as the hand-held device to control a ray or a cursor in virtual reality.

4.2 Experiment Design and Tasks

Four main factors should be considered in 3D target acquisition tasks [34, 38]: *Target Size*, *Target Depth*, *Density* and *Occlusion*. ***Target Size*** is one of the most influential factors for selection techniques in both 2D and 3D cases. Small targets are more apt to be difficult to select than large targets for some ray-casting techniques. ***Target Depth*** [13, 17, 39] causes many problems like distance perception

and access capability. In this experiment, we would like to first evaluate ray-casting with bubble mechanism in general cases, so we did not involve *Density* and *Occlusion* factors.

We designed a within-subject experiment with factors *Technique*, *Target Depth* and *Target Size*. There were three different depth ranges: near (0.2-0.7m, targets can be touched by hand), middle (1-5m, a room-level range) and far (10-30m, remote pointing). There were two different sizes of target diameter in each depth: large and small, totally six different scenes (see Table 1). We mainly focused on the comparison of techniques to figure out the pros and cons of the bubble mechanism, and the factors *Target Depth* and *Target Size* varied only to create scenes with different difficulties.

Table 1: Six scenes with different depth ranges and target sizes.

Scene	Depth Range	Target Size	Visual Size
Near-Large	0.2m-0.7m (near)	0.05m	4.1°-14.3°
Near-Small		0.02m	1.6°-5.7°
Middle-Large	1m-5m (middle)	0.25m	2.9°-14.3°
Middle-Small		0.10m	1.2°-5.7°
Far-Large	10m-30m (far)	1.00m	1.9°-5.7°
Far-Small		0.40m	0.8°-2.3°

In each of six scenes, ten spherical targets were presented in a random layout [28, 38] with the same size within the specified depth range. We guaranteed that there was no density or occlusion of targets, and all targets were easy to find from the user's view. The participant was required to select each target three times, a total of thirty selections per technique per scene. The selection order of targets was randomized for different participants in different scenes, but it remained the same for all techniques in the same scene for each participant to guarantee the same difficulty.

4.3 Evaluated Techniques

We chose seven 3D target acquisition techniques in the evaluation. All parameters of the techniques were well chosen for the best performance.

Bubble Cursor [38]: is the 3D extension of the original bubble cursor (Figure 4a). The user moves the controller to manipulate the cursor, while the cursor selects the nearest target included by the spherical bubble. We involve this technique because it represents a class of cursor-based techniques, and the bubble mechanism is also present. Since the access range of the cursor is limited around the body, Bubble Cursor is only evaluated in near scenes.

Go-Go [30]: allows the user to control a virtual hand to grab the target along the ray-casting direction (Figure 4b). The target falls into the range of the hand can be selected. The position of the virtual hand from the user's body depends on the position of the controller using a nonlinear mapping:

$$D_{vh} = \begin{cases} D_{ctrl} & if\ D_{ctrl} < D_0 \\ D_0 + k(D_{ctrl} - D_0)^2 & otherwise \end{cases}$$

where D_{vh} and D_{ctrl} denote the distance of the virtual hand and the controller from the user's body, D_0 is the threshold, and k is the nonlinear coefficient. The Go-Go technique represents a class of techniques that control an extensible 3D cursor along the ray. Due to the limit of the arm stretch, Go-Go is only evaluated in near and middle scenes.

Naive Ray [16]: is the original ray-casting metaphor (Figure 4c), which is the baseline of ray-casting techniques.

Heuristic Ray [7] uses spatial and temporal functions to compute a score for every target and select the one with the maximum score (Figure 4d):

$$s_t = s_{t-1}\lambda + \left(1 - \frac{\alpha(t)}{\beta}\right)(1-\lambda)$$

where s_t denotes the score at time t, $\alpha(t)$ denotes the angle between the ray and the center of the target at time t, β is the threshold, and λ is the time decay weight. There is also a curve target indicator connected to the selected target. Heuristic Ray represents a class of heuristic ray-casting techniques with an implicit scoring function to determine the optimal target.

Quad Cone [18, 21]: is a cone-shaped ray-casting technique with a quartered menu for disambiguation. The user first moves the cone to include the desired target and presses the selection button (Figure 4e). If there is only one target inside the cone, the selection will be finished, otherwise, the user needs to specify the desired target in the second step. All targets included by the cone are evenly distributed among four quadrants of the quartered menu in the second step (Figure 4f), and the user progressively eliminates undesired targets by repeatedly pointing the quadrant which contained the desired target, reducing the number of targets each time until there is only one target left. Quad Cone represents a class of two-step techniques that allow multi-selection and disambiguation.

Bubble Ray: indicates the ray-casting technique with bubble mechanism. According to the distance definition, we have two variants: **BubbleRay-E** (with Euclidean distance) and **BubbleRay-A** (with angular distance).

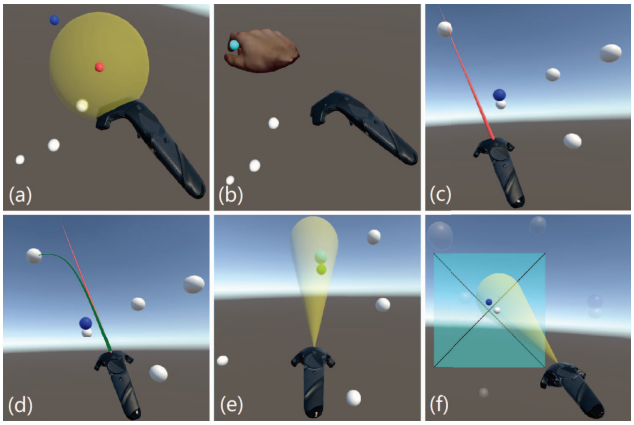

Figure 4: Evaluated techniques: (a) 3D Bubble Cursor. (b) Go-Go. (c) Naive Ray. (d) Heuristic Ray. (e)(f) Quad Cone: multi-selection with the cone; disambiguation on the quartered menu.

4.4 Procedure

At first, we introduced the goal of the experiment and guided the participant to use the device. The participant was required to sit and not move his or her body when wearing the head-mounted display. Next, the participant saw a sample scene in virtual reality. The desired target was highlighted in blue, and other targets were in white. The desired target became indigo when it was selected by the technique. The participant needed to press the trigger button on the controller to confirm the selection. After that, either a tone sound played to indicate a successful selection or a beep sound played to indicate a failed selection. The participant needed to try again after he failed until making a successful selection. Then, we introduced all seven techniques, and the participant was required to familiarize these techniques. The experiment started after the participant responded he had mastered all techniques.

The participant was required to use seven techniques in random order in order to counterbalance. With each technique, the participant was required to finish tasks in six scenes in the order of Table 1. In each scene, the participant had a warm-up phase to ensure he could see and select every target in the scene with the technique. Then, the participant was required to finish the task as fast as possible on the

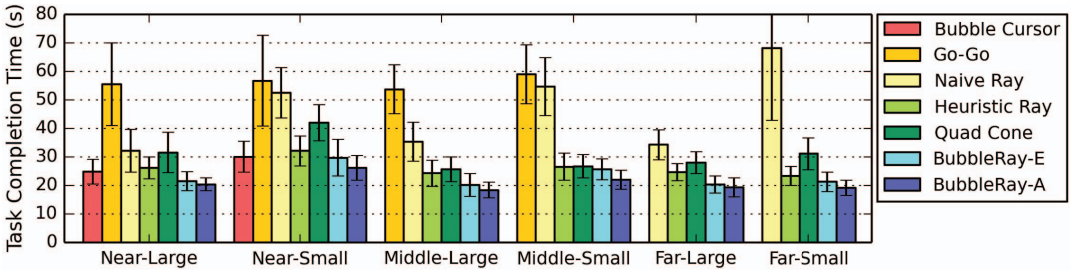

Figure 5: Task completion time of every technique in every scene. The error bar represents the standard deviation.

Table 2: Task completion time, error rate and moving distance of every technique.

Techniques	Bubble Cursor	Go-Go	Naive Ray	Heuristic Ray	Quad Cone	BubbleRay-E	BubbleRay-A
Completion Time (s)	27.44±5.47	56.23±12.43	46.17±18.02	26.20±4.97	30.85±7.52	23.12±5.29	20.90±4.05
Error rate (%)	0.97±2.50	15.00±11.02	23.47±19.33	4.35±4.43	0.39±0.39	2.73±4.04	1.76±3.06
Moving Distance (m)	14.34±2.61	8.56±2.01	5.13±1.59	4.89±1.59	6.83±2.00	5.38±2.37	4.99±2.00

premise of making no failed selection. After finishing the six scenes of one technique, the participant was required to take a break. The whole experiment took about one hour. After the experiment, the participant was required to fill a questionnaire based on a 7-point Likert scale to give subjective scores for all techniques.

4.5 Results

4.5.1 Task Completion Time

Task completion time of every technique in every scene was shown in Figure 5. Two Bubble Rays had good performances on task completion time in all scenes. An RM-ANOVA showed a significant effect of *Technique* on task completion time in near ($F_{6,66} = 52.4$, $p < 001$), middle ($F_{5,55} = 146$, $p < .001$) and far ($F_{4,44} = 61.1$, $p < .001$) scenes. Post hoc paired t-tests with Bonferroni correction showed BubbleRay-A significantly outperformed Bubble Cursor ($F_{1,11} = 18.6$, $p < .001$), Go-Go ($F_{1,11} = 161$, $p < .001$), Naive Ray ($F_{1,11} = 146$, $p < .001$), Heuristic Ray ($F_{1,11} = 60.1$, $p < .001$) and Quad Cone ($F_{1,11} = 319$, $p < .001$) on task completion time. BubbleRay-A reduced target completion time by at least 20% compared with other compared techniques. Between two Bubble Rays, A post hoc paired t-test with Bonferroni correction also showed BubbleRay-A significantly outperformed BubbleRay-E ($F_{1,11} = 36.9$, $p < .001$). This result supported our expectation that the angular definition was better to measure the distance between the ray and the target for the bubble mechanism.

4.5.2 Error Rate

We reported the error rate as the number of failed selections before one successful selection. Two Bubble Rays had relatively low error rates of less than 3% (Table 2). An RM-ANOVA showed a significant effect of *Technique* on error rates in near ($F_{6,66} = 19.2$, $p < .001$), middle ($F_{5,55} = 24.4$, $p < .001$) and far ($F_{4,44} = 40.8$, $p < .001$) scenes. Post hoc paired t-tests with Bonferroni correction showed BubbleRay-A had significantly lower error rate than Go-Go ($F_{1,11} = 29.8$, $p < .001$), Naive Ray ($F_{1,11} = 69.0$, $p < .001$) and Heuristic Ray ($F_{1,11} = 11.0$, $p = .007$), but no significant difference between BubbleRay-A and Bubble Cursor ($F_{1,11} = 2.27$, $p = .16$), Quad Cone ($F_{1,11} = 7.99$, $p = .05$) and BubbleRay-E ($F_{1,11} = 4.58$, $p = .06$).

4.5.3 Moving Distance

Moving distance [22, 32] of the controller reflects the difficulty of finding a good location and the fatigue of the user's hand. BubbleRay-A had the least moving distance (Table 2). An RM-ANOVA showed a significant effect of *Technique* on moving distance in near ($F_{1,11} = 81.4$, $p < .001$), middle ($F_{5,55} = 46.4$, $p < .001$) and far ($F_{4,44} = 29.4$, $p < .001$) scenes. Post hoc paired t-tests showed BubbleRay-A had significantly less moving distance than Bubble Cursor ($F_{1,11} = 190$, $p < .001$), Go-Go ($F_{1,11} = 46.7$, $p < .001$), Quad Cone ($F_{1,11} = 62.4$, $p < .001$) and BubbleRay-E ($F_{1,11} = 27.3$, $p < .001$), but no significant difference between BubbleRay-A and Naive Ray ($F_{1,11} = 0.272$, $p = .61$) or Heuristic Ray ($F_{1,11} = 0.238$, $p = .64$).

4.5.4 Subjective Feedback

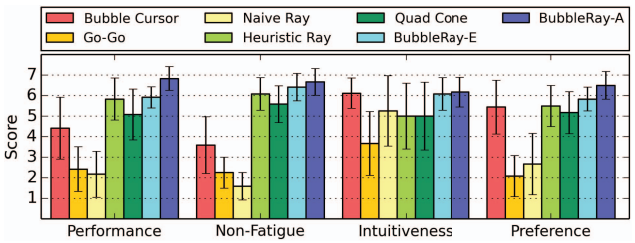

Figure 6: Subjective feedback from participants. The error bar represents the standard deviation. The score is from 1 to 7, the higher the better.

We required all participants to score between 1 to 7 from four perspectives. The technique with a higher score meant participants thought the technique did better in that perspective (Figure 6). A Friedman test showed a significant effect of *Technique* on Performance ($\chi^2(6) = 60.6$, $p < .001$), Non-Fatigue ($\chi^2(6) = 62.0$, $p < .001$), Intuitiveness ($\chi^2(6) = 50.614$, $p < .001$) and Preference ($\chi^2(6) = 24.099$, $p < .001$).

Performance reflects how easy and fast the user feels when selecting targets. Two Bubble Rays and Heuristic Ray were considered as techniques with higher performance.

Fatigue reflects the mental and physical cost for the user to make a successful selection. Two Bubble Rays, Heuristic Ray and Quad Cone were considered to cause less fatigue in target acquisition.

Intuitiveness reflects whether the user needs a learning and adaption process to achieve better use. Two Bubble Rays were considered more intuitive than other ray-casting techniques, and Bubble Cursor was considered intuitive in selecting near targets. This result showed

participants felt that the bubble mechanism improved the perception in target acquisition and made the selection more explicit.

Participants were also required to give an overall score for every technique to show their preferences. BubbleRay-A was shown to be preferred by participants, which proved ray-casting with bubble mechanism had good user experience.

In addition, some of the participants responded that the visual feedback of Bubble Rays did not match their perceptions in near scenes due to hand-eye parallax. They might prefer Bubble Cursor to select near targets although their hands had to move a longer distance.

4.6 Discussion

All metrics consistently showed BubbleRay-A (ray-casting with bubble mechanism using angular distance definition) performed well in target acquisition tasks. The results indicated the bubble mechanism could improve ray-casting techniques on the speed and accuracy of selection and reduce fatigue by expanding the effective width of targets. Also, the angular definition of distance was shown a little better than the Euclidean definition.

From another point of view, we did not study the effects of *Target Size* and *Target Depth* in detail since the finding was really trivial: smaller targets became more difficult to select for Naive Ray, and the depth had few effects on ray-casting techniques (RM-ANOVA: $F_{2,22} = 3.58$, $p = .45$). In other words, Heuristic Ray, Quad Cone and two Bubble Rays all had stable performances in every scene no matter how target depth and size changed. The only factor which influenced their performances was how targets were arranged in the scene (layout). In the next section, we will further investigate the effect of the difficulty of layouts.

5 Experiment 2: Evaluation in Extreme Cases

The layout of targets will influence the performance of techniques even the user's strategy using techniques. In this experiment, we will build scenes with extreme layouts to further evaluate ray-casting with bubble mechanism and investigate the benefits and limitations of the bubble mechanism for ray-casting in complicated tasks.

5.1 Scenes and Tasks

As we mentioned in Experiment 1, *Density* and *Occlusion* are the other two main factors in 3D target acquisition tasks besides *Target Size* and *Target Depth*. ***Density*** [4, 38, 41] of the environment influences the difficulty of distinguishing the desired target from its neighbors. The user has to provide more effort to select a target because of the less tolerance of the imprecision. ***Occlusion*** [13, 32, 38] means one target is partially or entirely hidden behind another target from the user's view. Occlusion may cause ambiguity in depths for some techniques. Considering these factors, we designed three extreme scenes for evaluation:

High Density (Figure 7a): 7×7 targets with the size of 0.1m form a square array at the depth of 5m. Two adjacent targets have a margin of 0.02m. This scene has no occlusion but density problems. There are 5×5 inner targets (targets not at the boundary) regarded as selection candidates, for the reason of ensuring there are four targets immediately adjacent to each candidate to make each selection fall into the dense case. The participant was required to select every candidate once, twenty-five selections in total.

High Occlusion (Figure 7b): An obstacle target locates exactly in front of the participant with the size of 0.75m at the depth of 5m. Four occluded targets form a rhombic layout and locate 0.8m behind the obstacle, so only a crescent-shaped part of each occluded target with an angle of 2° can be seen by the user. This scene has no density but occlusion problems. The participant was required to select every occluded target six times, twenty-four selections in total.

Density & Occlusion (Figure 7c): 5×5×5 targets with the size of 0.25m form a cubic layout [32, 40]. The center of the cubic layout is 3.5m in depth. Two adjacent targets have a margin of 0.25m. This scene has both density and occlusion problems. Targets at the cube boundary are not considered as selection candidates for the same reason as High Density scene. Targets located on the third row and third column are fully occluded, which are not considered as the candidates either. The participant was required to select every candidate once, twenty-four selections in total.

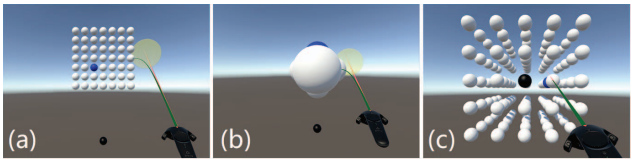

Figure 7: Three extreme scenes of Experiment 2 from the participant's view: (a) High Density. (b) High Occlusion. (c) Density & Occlusion. The black sphere indicates the reset target.

5.2 Participants, Experiment Design and Procedure

We recruited twelve participants (one female and eleven males, aged 21-25) from the university campus. All participants were right-handed. Five of them had participated in Experiment 1.

A within-subject design was used with the only factor *Technique*. Five techniques (Naive Ray, Heuristic Ray, Quad Cone, BubbleRay-E and BubbleRay-A) were evaluated in this experiment.

To control the difficulty of every selection, we placed a black target in each scene as a reset target (Figure 7). We required the participant moving back to the reset target before every selection to ensure every selection fell into the extreme case. That is, after a successful selection, the participant had to pass through the reset target, then he or she was allowed to make the next selection. The participant did not need to select or calibrate the reset target.

The participant was required to use five techniques in random order for the sake of counterbalancing. With each technique, the participant was required to finish the tasks of three scenes in random order. The selection order of targets was randomized for different participants in different scenes, but it remained the same for all techniques in the same scene for each participant to guarantee the same difficulty. The requirement of tasks was similar to that of Experiment 1, except the participant had to pass through the reset target before every selection. The whole experiment took about forty-five minutes.

5.3 Results

5.3.1 High Density Scene

Figure 8 showed the task completion time. Heuristic Ray and two Bubble Rays had less task completion time in this scene. An RM-ANOVA showed a significant effect of *Technique* on task completion time ($F_{4,44} = 28.0$, $p < .001$). Post hoc paired t-tests showed BubbleRay-A had significantly less task completion time than Naive Ray ($F_{1,11} = 8.89$, $p = .012$) and Quad Cone ($F_{1,11} = 51.5$, $p < .001$).

Figure 8 also showed the error rates. An RM-ANOVA showed a significant effect of *Technique* on error rates ($F_{4,44} = 9.17$, $p < .001$). Post hoc paired t-tests showed BubbleRay-A had significantly higher error rate than Heuristic Ray ($F_{1,11} = 15.0$, $p = .003$) and Quad Cone ($F_{1,11} = 20.6$, $p < .001$).

We expected Bubble Rays would degrade to Naive Ray in a highly dense scene since there was not too much empty space between targets, which made the effective width of targets (0.11m) only 10% greater than the actual width (0.1m). Due to the hand tremor problem, the error rate of Bubble Rays was also as high as that of

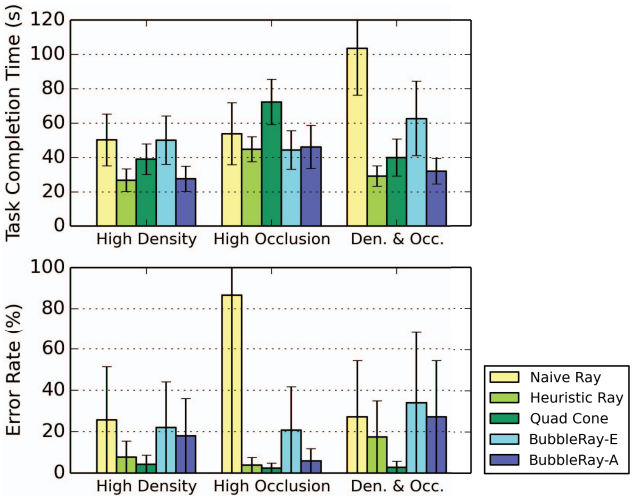

Figure 8: Task completion time and error rate for every technique in every scene. The error bar represents the standard deviation.

Naive Ray. However, the temporal model of Heuristic Ray provided the "sticky" feature [11] to avoid failed selections caused by the hand tremor.

5.3.2 High Occlusion Scene

An RM-ANOVA showed a significant effect of *Technique* on task completion time ($F_{4,44} = 73.2$, $p < .001$). Post hoc paired t-tests with Bonferroni correction showed BubbleRay-A had significantly less task completion time than Naive Ray ($F_{1,11} = 134$, $p < .001$), Quad Cone ($F_{1,11} = 8.51$, $p = .014$) and BubbleRay-E ($F_{1,11} = 47.9$, $p < .001$).

An RM-ANOVA also showed a significant effect of *Technique* on error rates ($F_{4,44} = 102$, $p < .001$). Post hoc paired t-tests showed BubbleRay-A had significantly lower error rate than Naive Ray ($F_{1,11} = 224$, $p < .001$) and BubbleRay-E ($F_{1,11} = 11.1$, $p = .007$).

The drawback of Euclidean distance definition was clearly shown in this scene. The front obstacle in this scene would always be the nearest target according to the Euclidean distance, so the participant had to make more effort to directly intersect the occluded target as Naive Ray. However, angular distance definition did not have this problem. The space around occluded targets would still contribute to the effective width to make the targets easy to select.

5.3.3 Density & Occlusion Scene

An RM-ANOVA showed a significant effect of *Technique* ($F_{4,44} = 6.37$, $p < .001$) on task completion time. Post hoc paired t-tests with Bonferroni correction showed BubbleRay-A had significantly less task completion time than Quad Cone ($F_{1,11} = 28.6$, $p < .001$).

An RM-ANOVA also showed a significant effect of *Technique* on error rates ($F_{4,44} = 6.85$, $p < .001$). Post hoc paired t-tests with Bonferroni correction showed BubbleRay-A had significantly higher error rate than Quad Cone ($F_{1,11} = 16.9$, $p = .002$).

In this scene, Quad Cone had the highest task completion time with the lowest error rate due to its additional disambiguation phase, while other techniques had similar performance with very high error rates. This result showed that ray-casting techniques should be associated with extra disambiguation steps or additional modalities (e.g., Depth Ray [13]) in scenes with high density and occlusion.

6 Discussion

The results in Experiment 1 showed Bubble Rays had a high performance of selection in simple tasks. Bubble Rays expanded the effective width of targets and reduced the moving distance, which made targets easy for selection. Users also liked the explicit visualization of the bubble which made the selection intuitive. However, Bubble Rays had the limitation that they did not perform well in tasks with dense targets because the effective width was compressed. Although their task completion time was not worse than other techniques, the error rate was really high. According to the response from users, they preferred to spend more time to achieve a higher success rate due to the high cost of re-try.

We find the angular definition of the distance is better than the Euclidean definition for Bubble Ray. BubbleRay-A can be interpreted as the selection on a Voronoi diagram [12] of a 2D sphere, which better matches users' fields of view. On the other hand, BubbleRay-E makes the selection on a 3D Voronoi diagram, which causes confusion along with the depth. However, there is still an obvious hand-eye parallax when selecting near targets when using BubbleRay-A. A more comprehensive model involving hand positions can be considered to improve Bubble Ray, and this will be our future work.

There are other limitations in our work. First, we did not allow the participants to walk around to seek for a better view in the experiments. In real VR scenarios, body movements should be also considered. Second, we did not compare more ray-casting techniques with other modalities, for example, Depth Ray [13] or RayCursor [1] with the touchpad on the controller. In extreme cases, these additional modalities may help for disambiguation. Third, our analysis can be improved by investigating the throughput of techniques and modeling with Fitts' law [10] to have a better view of the speed-accuracy tradeoff. Fourth, Bubble Ray ensures there must be one target selected, because of which the intention of selection should be inferred in real use to avoid false triggering. We propose several potential improvements in the following as our future work.

6.1 Speed/Surrounding-Dependent Sticky Ray

Results in Experiment 2 showed BubbleRay-A had a high error rate in dense cases, while Heuristic Ray with implicit sticky feature had a relatively low error rate. This indicates that the sticky mechanism works in dense cases. DynaSpot [5] has shown the idea of degrading the bubble cursor into the original point cursor when the cursor slows down. We can also redesign Bubble Ray in this way: when the speed of controller is low (or the surrounding of the ray is dense), Bubble Ray becomes original ray-casting with the sticky mechanism which sticks the nearest target to avoid hand tremor; otherwise, it still retains the bubble mechanism with high performance.

6.2 Gaze-Filtered Bubble Ray

The bubble mechanism guarantees exactly one target is selected at any time. However, the user may select an unintentional target far away from the ray when the environment is sparse. It can be easily solved by setting a threshold of maximum angular distance between the ray and the target. Considering the hand-eye parallax, we can also involve the gaze direction as a filter to help determine the target. We can hypothesize that the user will look at the target when he wants to select, so targets out of the range of the user's view will not be considered as the desired target. With this filtering, the bubble mechanism will not select the target out of the user's attention.

6.3 Bubble Ray + Bubble Cursor

From our subjective feedback in Experiment 1, Bubble Cursor was more intuitive and preferred in near scenes. A lot of virtual reality games also require the user to directly touch on the target to increase engagement. One idea is to combine Bubble Cursor and Bubble Ray together. When the user selects targets around him, the controller becomes the 3D bubble cursor to provide direct and enjoyable touch; when the user tries to select a distant target, the controller becomes

the ray-casting with bubble mechanism to provide fast and robust target acquisition.

7 Conclusion

In this work, we iteratively refine the visual feedback of the bubble mechanism for ray-casting from users' comments and render a disc-shaped bubble tangent to the target as the final design. Our two experiments show BubbleRay-A has high performance with less selection time and better user experience in both general and extreme cases. Also, the angular distance is shown better than the Euclidean distance for the bubble mechanism. Based on the findings from experiments, we propose some variants and improvements of Bubble Ray to make it more practical in real use.

Acknowledgments

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61521002, No. 61672314, and also by Beijing Key Lab of Networked Multimedia.

References

[1] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI 19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300331

[2] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, pp. 35–ff. ACM, New York, NY, USA, 1997. doi: 10.1145/253284.253301

[3] D. A. Bowman, D. B. Johnson, and L. F. Hodges. Testbed evaluation of virtual environment interaction techniques. *Presence: Teleoperators and Virtual Environments*, 10(1):75–95, 2001. doi: 10.1162/105474601750182333

[4] J. Cashion, C. Wingrave, and J. J. L. Jr. Dense and dynamic 3d selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):634–642, April 2012. doi: 10.1109/TVCG.2012.40

[5] O. Chapuis, J.-B. Labrune, and E. Pietriga. Dynaspot: Speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1391–1400. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518911

[6] N. T. Dang, H. H. L. H.-H. Le, and M. Tavanti. Visualization and interaction on flight trajectory in a 3d stereoscopic environment. In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, vol. 2, pp. 9.A.5–91–10 vol.2, Oct 2003. doi: 10.1109/DASC.2003.1245905

[7] G. De Haan, M. Koutek, and F. H. Post. Intenselect: Using dynamic object rating for assisting 3d object selection. In *IPT/EGVE*, pp. 201–209. Citeseer, 2005. doi: 10.2312/EGVE/IPT_EGVE2005/201-209

[8] H. G. Debarba, J. G. Grandi, A. Maciel, L. Nedel, and R. Boulic. *Disambiguation Canvas: A Precise Selection Technique for Virtual Environments*, pp. 388–405. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-40477-1_24

[9] N. Elmqvist and J.-D. Fekete. Semantic pointing for object picking in complex 3d environments. In *Proceedings of Graphics Interface 2008*, GI '08, pp. 243–250. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 2008. doi: 10.1145/1375714.1375755

[10] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954. doi: 10.1037//0096-3445.121.3.262

[11] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 95–96. ACM, New York, NY, USA, 1996. doi: 10.1145/237091.237105

[12] T. Grossman and R. Balakrishnan. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pp. 281–290. ACM, New York, NY, USA, 2005. doi: 10.1145/1054972.1055012

[13] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pp. 3–12. ACM, New York, NY, USA, 2006. doi: 10.1145/1166253.1166257

[14] M. Guillon, F. Leitner, and L. Nigay. Static voronoi-based target expansion technique for distant pointing. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pp. 41–48. ACM, New York, NY, USA, 2014. doi: 10.1145/2598153.2598178

[15] M. Guillon, F. Leitner, and L. Nigay. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 2777–2786. ACM, New York, NY, USA, 2015. doi: 10.1145/2702123.2702375

[16] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pp. 213–222. ACM, New York, NY, USA, 1994. doi: 10.1145/192426.192501

[17] I. Janzen, V. K. Rajendran, and K. S. Booth. Modeling the impact of depth on pointing performance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 188–199. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858244

[18] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3d selection by progressive refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–74, March 2011. doi: 10.1109/3DUI.2011.5759219

[19] R. Kopper, D. A. Bowman, M. G. Silva, and R. P. McMahan. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies*, 68(10):603 – 615, 2010. doi: 10.1016/j.ijhcs.2010.05.001

[20] A. Kulshreshth and J. J. LaViola, Jr. Dynamic stereoscopic 3d parameter adjustment for enhanced depth discrimination. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 177–187. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858078

[21] J. Liang and M. Green. Geometric modeling using six degrees of freedom input devices. In *3rd Int'l Conference on CAD and Computer Graphics*, pp. 217–222. Citeseer, 1993.

[22] J. Looser, M. Billinghurst, R. Grasset, and A. Cockburn. An evaluation of virtual lenses for object selection in augmented reality. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, pp. 203–210. ACM, New York, NY, USA, 2007. doi: 10.1145/1321261.1321297

[23] I. S. MacKenzie and W. Buxton. Extending fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pp. 219–226. ACM, New York, NY, USA, 1992. doi: 10.1145/142750.142794

[24] M. Mine et al. Virtual environment interaction techniques. *UNC Chapel Hill computer science technical report TR95-018*, pp. 507248–2, 1995.

[25] M. E. Mott and J. O. Wobbrock. Beating the bubble: Using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 733–742. ACM, New York, NY, USA, 2014. doi: 10.1145/2556288.2557410

[26] D. R. Olsen, Jr. and T. Nielsen. Laser pointer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pp. 17–22. ACM, New York, NY, USA, 2001. doi: 10.1145/365024.365030

[27] A. Olwal, H. Benko, and S. Feiner. Senseshapes: using statistical geometry for object selection in a multimodal augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 300–301, Oct 2003. doi: 10.1109/ISMAR.2003.1240730

[28] S. Park, S. Kim, and J. Park. Select ahead: Efficient object selection

technique using the tendency of recent cursor movements. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction*, APCHI '12, pp. 51–58. ACM, New York, NY, USA, 2012. doi: 10.1145/2350046.2350060

[29] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, pp. 39–ff. ACM, New York, NY, USA, 1997. doi: 10.1145/253284.253303

[30] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 79–80. ACM, New York, NY, USA, 1996. doi: 10.1145/237091.237102

[31] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum*, 17(3):41–52, 1998. doi: 10.1111/1467-8659.00252

[32] Z. Serrar, N. Elmarzouqi, Z. Jarir, and J.-C. Lapayre. Evaluation of disambiguation mechanisms of object-based selection in virtual environment: Which performances and features to support "pick out"? In *Proceedings of the XV International Conference on Human Computer Interaction*, Interacción '14, pp. 29:1–29:8. ACM, New York, NY, USA, 2014. doi: 10.1145/2662253.2662282

[33] A. Steed. Towards a general model for selection in virtual environments. In *3D User Interfaces (3DUI'06)*, pp. 103–110, March 2006. doi: 10.1109/VR.2006.134

[34] A. Steed and C. Parker. 3d selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th International Immersive Projection Technology Workshop*, pp. 13–14, 2004.

[35] F. Steinicke, T. Ropinski, and K. Hinrichs. *Object Selection in Virtual Environments Using An Improved Virtual Pointer Metaphor*, pp. 320–326. Springer Netherlands, Dordrecht, 2006. doi: 10.1007/1-4020-4179-9_46

[36] A. Toet. Visual comfort of binocular and 3d displays. *Displays*, 25(2):99–108, 2004. doi: 10.1016/j.displa.2004.07.004

[37] E. Tse, M. Hancock, and S. Greenberg. Speech-filtered bubble ray: Improving target acquisition on display walls. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, ICMI '07, pp. 307–314. ACM, New York, NY, USA, 2007. doi: 10.1145/1322192.1322245

[38] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *2007 IEEE Symposium on 3D User Interfaces*, March 2007. doi: 10.1109/3DUI.2007.340783

[39] G. Wang, M. J. McGuffin, F. Bérard, and J. R. Cooperstock. Pop-up depth views for improving 3d target acquisition. In *Proceedings of Graphics Interface 2011*, GI '11, pp. 41–48. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2011.

[40] C. A. Wingrave, R. Tintner, B. N. Walker, D. A. Bowman, and L. F. Hodges. Exploring individual differences in raybased selection: strategies and traits. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pp. 163–170, March 2005. doi: 10.1109/VR.2005.1492770

[41] J. Wonner, J. Grosjean, A. Capobianco, and D. Bechmann. Starfish: A selection technique for dense virtual environments. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, VRST '12, pp. 101–104. ACM, New York, NY, USA, 2012. doi: 10.1145/2407336.2407356

[42] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pp. 266–271. ACM, New York, NY, USA, 1997. doi: 10.1145/258549.258724

[43] S. Zhai, W. Buxton, and P. Milgram. The "silk cursor";: Investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pp. 459–464. ACM, New York, NY, USA, 1994. doi: 10.1145/191666.191822