MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments

William Delamare, Maxime Daniel, Khalad Hasan

To cite this version:

William Delamare, Maxime Daniel, Khalad Hasan. MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments. CHI '22: CHI Conference on Human Factors in Computing Systems, Apr 2022, New Orleans LA, United States. pp.1-6, 10.1145/3491101.3519692 . hal-04008734

HAL Id: hal-04008734
https://hal.inria.fr/hal-04008734

Submitted on 28 Feb 2023

MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments

William Delamare
Univ. Bordeaux, ESTIA INSTITUTE OF TECHNOLOGY
Bidart, France
w.delamare@estia.fr

Maxime Daniel
Univ. Bordeaux, ESTIA INSTITUTE OF TECHNOLOGY
Bidart, France
m.daniel@estia.fr

Khalad Hasan
University of British Columbia
Kelowna, BC, Canada
khalad.hasan@ubc.ca

ABSTRACT

In this work, we propose MultiFingerBubble, a new variation of the 3D Bubble Cursor. The 3D Bubble Cursor is sensitive to distractors in dense environments: the volume selection resizes to snap-to nearby targets. To prevent the cursor to constantly re-snap to neighboring targets, MultiFingerBubble includes multiple targets in the volume selection, and hence increases the targets effective width. Each target in the volume selection is associated with a specific finger. Users can then select a target by flexing its corresponding finger. We report on a controlled in-lab experiment to explore various design options regarding the number of fingers to use, and the target-to-finger mapping and its visualization. Our study results suggest that MultiFingerBubble is best used with three fingers and colored lines to reveal the mapping between targets and fingers.

CCS CONCEPTS

• **Human-centered computing → Pointing**; *Laboratory experiments.*

KEYWORDS

3D selection, Virtual Hand, Bubble Cursor, Pointing, MultiFingerBubble, Dense, Virtual Reality, Augmented Reality, Mixed Reality

ACM Reference Format:
William Delamare, Maxime Daniel, and Khalad Hasan. 2022. MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3491101.3519692

1 INTRODUCTION

The extensive body of work about target selection in 3D environments [2, 8, 9] can be classified into two categories [2, 9]: the *Virtual Hand* (i.e., selection by direct or indirect touching) and the *Ray Casting* (i.e. selection by pointing) metaphors. Both categories face difficulties when dealing in dense environments because of limited motor accuracy.

We focus on the selection of 3D objects in small-scaled environments, such as a smart-home allowing users to select and control appliances via a small-scaled 3D home replica displayed on demand under the user's hand via Augmented Reality, or a cockpit in immersive Virtual Reality allowing users to select a drone via a small-scaled 3D map of drones near user's hand. More specifically, we consider the use of a *Virtual Hand* solution, namely the 3D Bubble Cursor [29]. The 3D Bubble Cursor suffers two limitations. Indeed, as a Virtual Hand metaphor, it offers a limited range after which users cannot attain targets, even with custom gain allowing further reach [24]. In addition, automatically selecting the nearest target is highly sensitive to nearby distractors [3, 15, 18]. Thus, even small movements in dense environments can create incessant disturbing effect with the technique constantly snapping to nearby targets. Very few improvements have been proposed since the appearance of the 3D Bubble Cursor technique [22]. We hence undertake the task of proposing a solution to improve the 3D Bubble Cursor regarding two common issues, namely (i) the re-snap effect in dense environment, and (ii) the limited reaching space. To do so, we present the design of *MultiFingerBubble.*

MultiFingerBubble allows users to control a semi-transparent sphere by mapping the hand position onto the sphere position. Like the 3D Bubble Cursor [29], the sphere resizes its radius as to always select the closest target. However, the sphere can contain multiple targets - up to four. Each target is linked to a specific finger. Users can then select the desired target by flexing the corresponding finger. Thus, the more targets are included in the sphere, the larger the effective width, i.e. the target size in motor space instead of the visual space. We also consider the case of multiple clusters of targets. For instance, a virtual cockpit could display a 3D map of aerial drones on the left side, and a 3D map or terrestrial robots on the right side. We hence propose a sphere casting [11] with the palm to let users decide in which cluster the sphere should appear without moving through empty spaces in-between clusters.

We report on a preliminary user study to explore critical design options. One fundamental design choice concerns the number of fingers, i.e. the number of targets included in the volume selection. We hence test MultiFingerBubble with 1 finger - like the original 3D Bubble Cursor - 2, 3, and 4 fingers. We also explore finger-target mappings. The mapping can be based on the actual targets' layout, i.e., left-to-right mapping. However, changes in the volume selection such as targets exiting/entering the volume would change the mapping of every target, even the ones already in the volume. We hence also explore a stable mapping, i.e. only targets entering the volume are assigned to new fingers. Lastly, we explore two visualizations to provide information about the finger-target mapping: colored icons or colored lines from fingertips to targets. Results

---

CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9156-6/22/04…$15.00
https://doi.org/10.1145/3491101.3519692

show that MultiFingerBubble is best used with three fingers, and with lines displaying the stable mapping between targets and fingers. This indicates an improvement compared to the original 3D Bubble Cursor, i.e., MultiFingerBubble with one finger.

2 RELATED WORK

Our research builds on 3D target selection in a virtual environment. We also review cursor enhancement techniques and conclude with a brief review of finger based selection techniques that we leverage to design MultiFingerBubble.

2.1 3D Target Selection

We briefly review the two approches (i) using a virtual hand, and (ii) using a virtual pointer used for target selection in 3D environments. With the virtual hand metaphor, a user selects a target by touching targets with a virtual representation of their real hands, which uses a linear [26] or non-linear [24] mapping between the real hand and the virtual hand. This metaphor offers an intuitive interaction paradigm in 3D environments for target selection [2]. However, virtual hand techniques are limited when selecting out-of-reach objects. On the contrary, virtual pointer techniques use a virtual ray adapted for distant selection. The ray emits from users' hands – allowing them to control the origin and trajectory of the ray [20, 27, 30]. Researchers compared performances of these two target selection metaphors and showed that virtual pointers (e.g., ray casting) outperforms the virtual hand metaphor [6, 26].

Prior work also explored alternate versions of ray casting, such as Head-Based RayCasting, where the ray emits from users' heads and moves based on the head orientation [9, 17, 25]. In addition, researchers explored other ray casting based techniques such as Depth Ray [20, 27, 30] to assist users with selecting occluded items in virtual environments.

2.2 Cursor Enhancement Techniques

Pointing performance can be generally modelled with Fitts' law [13]: $T = a + blog_2(A/W + 1)$ where the movement time T depends on the target width W and distance (or amplitude) A, and a and b are empirically determined constants. Researchers explored different enhanced cursor techniques by changing these parameters. For instance, the Bubble Cursor [14] enhances the 2D area cursor technique by dynamically adjusting the size of the cursor to always capture the closest target, and hence minimizes the travel distance amplitude A. This cursor technique is widely studied in sparse and dense target environments and has shown to be an effective technique for 2D target selection. Inspired by the results, researchers designed and explored different cursor enhancement techniques adapted to 3D environments.

Vanacken et al. [29] proposes the 3D Bubble Cursor that dynamically resizes the volume selection to contain the closest target within the cursor's boundaries. Their experiment reveals that the 3D bubble cursor had competitive performances compared with the standard point cursor and ray-casting technique. Similarly, for the virtual pointer metaphor, Steinicke et al. [28] proposed Sticky-Ray, a bendable ray that points to the last highlighted select target unless the ray hits another selectable target. Researchers also compared the performance of the 3D Bubble Cursor with Expanding Targets [1], and the Sticky Ray [28] where results revealed that RayCasting is significantly slower than the 3D Bubble Cursor, and that participants preferred the 3D Bubble Cursor over the other techniques. Previous work also combined ray-casting with the bubble mechanism. For instance, RayCursor [6] proposes a cursor along the ray. Users can control the cursor's depth along the ray via the Vive controller trackpad. The target closest to the cursor can then be selected via the trigger button. In a recent work, Lu et al. [19] explored BubbleRay, a ray-casting technique with a bubble mechanism. With BubbleRay, the ray always contains the closest target, allowing users to select targets without accurate aim. To evaluate the BubbleRay, authors consider five other selection techniques: a naive ray casting, the 3D Bubble Cursor, the Go-Go technique [24], Intenselect [12], and Quad Cone [15]. Results showed the bubble mechanism improves ray-casting regarding both performances and preferences compared to all other techniques.

However, such snap-to feature is sensitive to distractors in dense environments [3, 15, 18]. Indeed, the snap-to feature is highly responsive to hand movements creating constant resize of the sphere to snap-to nearby targets. The technique is then equivalent to using a point-cursor [21], and the re-snap effect might be distracting [7]. In this work, we undertake the challenge to improve the 'snap-to' enhancement technique of the Virtual Hand metaphor.

2.3 Finger-Based Selection Techniques

Finger-based selection techniques have been used to select items, especially with menu-based systems. For instance, the two-handed TULIP menu displays items on fingers in virtual environments [10]. Users can then select items and navigate in the menu hierarchy by pinching the corresponding finger and the thumb. Bailly et al. [4] proposed Finger-Count menus to use with multitouch surfaces. Finger-Count leverages the information of the number of fingers of each hand in contact with the surface to select an item. A similar idea has also been adopted with distant displays (e.g., Kinect [5]). The user study reveals that finger count requires more mental demand than selection techniques such as marking menu [16]. Leveraging fingers for target selection has also been used in other contexts such as when interacting with head-mounted displays. For instance, in a recent work, Jordan et al. [23] proposed CountMarks, which leverages the number of fingers used on both hands to open a menu and trigger a menu items selection. CountMarks offers faster selections compared to marking menus [16]. Though fingers have been used to trigger selections, to the best of our knowledge, no prior work explored ways to leverage fingers in conjunction with a bubble cursor to facilitate target selections in dense 3D environments.

3 DESIGN EXPERIMENT

We design MultiFingerBubble by exploring design options related to the target-finger mapping.

3.1 MultiFingerBubble and Design Factors

Originally, the 3D Bubble Cursor allows users to control a semi-transparent sphere by mapping the hand position onto the sphere position. The sphere resizes its radius as to always select the closest target. The main benefits of the 3D Bubble Cursor (and its 2D origin

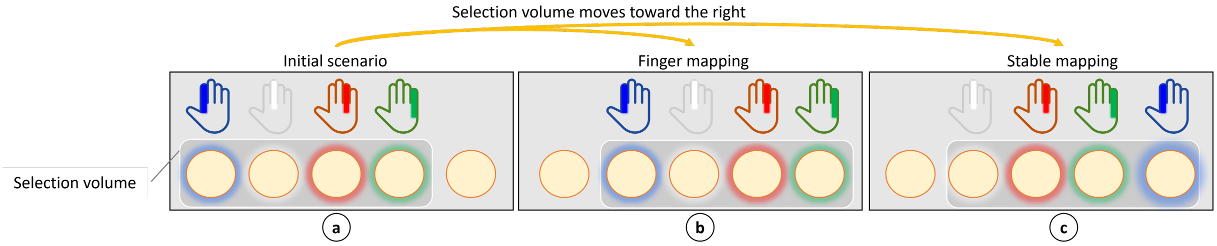

Figure 1: Illustrations of each mapping's principle. a) Original situation. Each target is associated to a specific finger and its color. This example portrays a finger mapping: targets in the volume selection are linked to the fingers corresponding to their horizontal ordering. b) With a finger mapping, if a target (the left one) is replaced by another (the right one) in the volume selection, all targets need to update their color. c) With a stable mapping, only the color of the target replaced in the volume selection is updated - from the leaving target to the entering target. Targets remaining in the volume selection keep their color.

version [14]) are twofold. First, the technique reduces unnecessary movement in empty areas by automatically resizing its radius. Second, the technique facilitates the selection of small targets by increasing its effective width, i.e. the target size in motor space instead of the visual space [29]. MultiFingerBubble adds two main differences to the original 3D Bubble Cursor. First, we provide a *sphere casting* mechanism to let users position the bubble in a specific cluster without unnecessary hand movements in empty spaces in-between clusters. Once the user aims at a cluster with her palm, the virtual sphere follows the hand movement with a 1:1 mapping. Second, we provide the possibility to select more than one target included in the semi-transparent sphere. We expect these changes to (i) provide faster selection by reducing unnecessary movements, and (ii) provide more accurate selection by reducing the negative impact of increased density on the snap-to feature of the bubble [3, 15, 18]. To select a specific target included in the volume selection, users perform a pinch gesture with the corresponding finger and their thumb. We focus on the multi-finger design aspects, not on the sphere casting from the palm. Thus, for the fine-tuning of MultiFingerBubble's design, we consider three factors, the *Number of fingers*, the *Visualization* and the *Mapping*.

3.1.1 Number of Fingers. We envision the use of four fingers to perform pinch gestures: The index, the middle, the ring, and the pinky. We hypothesize that the number of fingers creates a trade-off between (i) the increased stability of the volume selection's content, and (ii) the motor and cognitive efforts required to select a target.

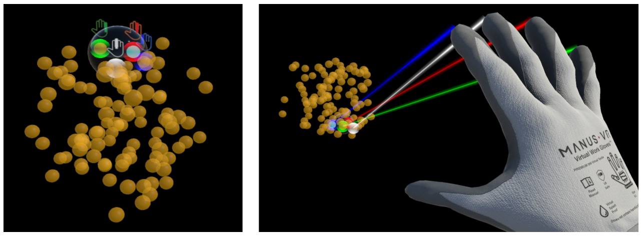

Figure 2: A stable mapping between 4 fingers and 4 targets in a cluster of 100 targets using Icons (left) and Lines (right) as visual cues.

Our goal is to find the number of fingers corresponding to the best stability/efforts trade-off. We consider the use of 1 (index finger) - like the original 3D Bubble Cursor -, 2 (index and middle fingers), 3 (index, middle and ring fingers), and 4 fingers (all fingers).

3.1.2 Visualization. We explore two visualizations to reveal the mapping between targets and fingers to users: lines and icons. Both visualizations build on the same color scheme to help users memorize the target-finger mapping: blue for index, white for middle, red for ring, and green for pinky. Colored lines start from the fingertips and end on the targets (Figure 2, right). Colored icons are positioned 3.5 cm above the corresponding targets and display a 3 cm×3 cm hand image with the corresponding finger filled with the corresponding color (Figure 2, left). For both visualizations, targets also display a colored halo, with the color matching the associated finger's color.

3.1.3 Mapping. We consider two finger-target mapping options: Following the fingers direction (from index to pinky), or stable. With the finger mapping option, colors are assigned according to the horizontal ordering of targets to coincide with the fingers ordering (from index to pinky). For instance with a right-handed user, the target on the left will be assigned to the index finger (blue color), the next target on the right to the middle finger (white color), etc. (Figure 1, a). While this intuitive mapping can help users to quickly determine the associated finger to validate a selection, the finger mapping can create last-second change of colors. For instance, if the user wants to select the red target (Figure 1, a), but moves her hand while starting to flex the ring finger, then some targets might enter while some others might exit the volume selection. In this case, every target has to update their color according to their new horizontal ordering and their newly associated fingers (Figure 1, b). On the figure, the red target (ring finger) becomes white (middle finger). Thus, while the target remains in the volume selection, the effect of the increased effective width is lost during last-second finger assignment. We hence explore another option with the opposite properties, i.e. without clear mapping rationale, but reducing risks of last-second color changes: a stable mapping. If a target exits the volume and a new one enters, only the impacted color (and hence finger assignment) switches target (Figure 1, c). In the previous scenario, the blue color is re-affected to the new target,

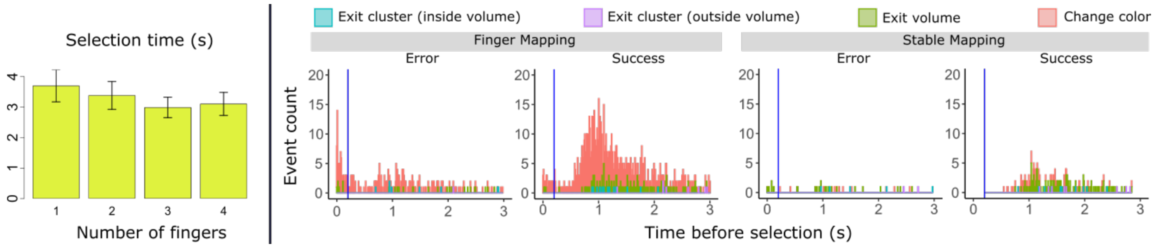

Figure 3: Left: Selection time in seconds according to the number of finger used. Right: Histogram of events happening right before a selection for both finger and stable mappings, in case of successful and erroneous selection. The blue vertical lines represent the threshold of interest at 200 ms before a selection.

but all other targets keep their original color. Note that the Finger mapping can let users choose the finger they desire to perform the selection by positioning the desired target in a specific horizontal position to match the desired finger. With the stable mapping, such control is not possible since colors and fingers are dynamically associated with entering / exiting targets.

3.2 Participants and Apparatus

We recruited 16 participants (7 females, 9 males), aged 20 to 54 (M=35, SD=11.8) from our local university but outside the computer science department. All participants were right-handed, and had none (4) to little (9) experience with Virtual Reality. Three participants had more experience due to gaming. We used a 3.8 GHz Intel Core i7 desktop computer and implemented the software with Unity 3D 2019.4. We used a Vive Pro system, with a marker attached to Manus Prime II Haptic Gloves[1] to track hand and finger movements.

The hardware could not accurately detect pinch gestures, i.e. actual contact between the thumb and the other fingers. We hence switched from pinching gestures to finger flexion resembling a midair finger tap.

3.3 Procedure

The experiment lasted between 1h and 1h30 per participant. All participants wore masks and disinfected their hands upon arrival due to the COVID-19 pandemic. After providing demographic data and signing a consent form, participants were presented the goal of the study. We then performed a short calibration of the Manus gloves to fit participants hands. The session was divided into two parts: one for each *Mapping*. Each part was divided into two sequences corresponding to *Visualization*. For each *Mapping* × *Visualization* condition, participants performed 3 blocks of 10 selections with 1, then 2, then 3, and then 4 fingers. The 3D scene was composed of two clusters approximately 1m in front of the user: one containing only the ‘start’ target, and another 20 cm × 20 cm × 20 cm cluster, 50 cm on the right, containing 100 targets. Targets were 2 cm semi-transparent yellow spheres. Although we do not thoroughly explore the sphere casting mechanism from the palm, we position the target cluster close to ‘start’ to establish if users can maintain their palm orientation while moving the bubble or flexing the fingers. We first calibrated the volume selection, a transparent sphere, so that its center was in the middle of the cluster in a comfortable position (i.e., elbow on the armrest of the experimental chair, and hand in midair). At the beginning of a trial, the software displayed only the ‘start’ target, approximately 1m in front of the participants. Participants had to validate the selection of ‘start’ for the system to display the cluster with the actual targets. The target to select was colored in cyan. Once a selection validated, the start target became yellow again and the cluster of targets disappeared. A sound indicated success or failure of the selection. In case of an incorrect selection, the trial was automatically repeated at the end of the sequence. Before each *Visualization* × *Mapping* × *Number of fingers* condition, participants had 10 practice trials. The targets position were randomized between each condition. In conditions with more than one finger, ‘start’ could be selected with any available finger. At the end of the session, participants were asked to rank *Visualization*, *Mapping*, and *Number of fingers* options.

3.4 Experimental Design

We used a within-participants design with *Visualization* (Lines, Icons) and *Mapping* (Finger, Stable) counter-balanced using a Latin-square design. For each condition, the *Number of fingers* (1 to 4) progressively increased: participants started with 1 finger, and finished with 4 fingers. Indeed, our goal is to establish the comfortable number of fingers to use with MultiFingerBubble. Directly starting with 3 or 4 fingers would hinder learning effects instead of revealing how an extra-finger impacts performances. For each finger condition, participants performed 3 blocks of 10 trials, i.e. a selection of a randomly highlighted target. We hence collected 2 *Visualizations* × 2 *Mappings* × 4 *Number of fingers* × 3 Blocks × 10 trials = 480 successful trials per participants, 7680 trials in total.

3.5 Results

Our dependent variables are the selection time (duration between ‘start’ selection and correct target selection) of successful trials, and success rate (ratio of incorrect selections and total number of selections). We perform our analysis with non-parametric tests (Friedman and Wilcoxon tests), with all post-hoc tests reporting Bonferroni corrected p-values. We report averaged values with their 95% confidence intervals (Mean, [CI low, CI high]) and effect sizes. We removed 42 trials (0.47%) due to double selection of ‘start’ (similar to a double click).

---

[1] https://www.manus-vr.com/haptic-gloves (last access: August 2021)

3.5.1 Selection Time. There is no effect of *Visualization* [W=45, Z=-1.19, p=0.25] or *Mapping* [W=75, Z=0.36, p=0.74] on selection time. However, there is a significant effect of *Number of fingers* [$\chi^2(3)$=19, p<.0001]: having 3 (M=2.99s, [2.66, 3.33]) or 4 (M=3.1s, [2.72, 3.48]) fingers allows faster selections than only 1 (M=3.7s, [3.17, 4.22]) or 2 (M=3.38s, [2.93, 3.83]) fingers with small to medium effect sizes (all p<0.01, all r between 0.2 and 0.4) (Figure 3, left).

3.5.2 Success Rate. We did not find any significant effect of *Visualization* [W=32, Z=-1.53, p=0.14] or *Mapping* [W=34, Z=-1.53, p=0.14]. There is an effect of *Number of fingers* [$\chi^2(3)$=13, p<.0001], with 3 fingers (M=85%, [80.4, 89.0]) leading to more accurate selections than 4 fingers (M=81%, [77.0, 85.4]) with a small effect size (p<0.05, r=0.25).

The overall success rate is around 83%. To better understand this relatively low percentage, we further explore the errors by distinguishing four events happening toward the end of the selection process (Figure 3, right): The bubble could jump to the 'start' cluster, i.e. the palm aims the 'start' target. This can happen while the target is inside the selection volume (e.g., wrist rotation while flexing a finger, Figure 3 blue), or when the target is not inside the selection volume (e.g., wrist rotation due to hand movements while moving the bubble, Figure 3 purple). Other errors can be related to the volume selection instead of the palm orientation. This can happen when the target exits the bubble (Figure 3 green), or lastly, when the target changes color while remaining in the volume selection (Figure 3 red). In case of unsuccessful trial, with a Finger mapping, most events correspond to targets changing colors (Figure 3, right). For the Stable mapping, events mostly involve targets exiting the volume selection. The same applies to successful trials, but in smaller proportions, especially when considering the last 200 ms (Figure 3, blue vertical line).

3.5.3 Preferences. While we did not find any quantitative differences regarding *Visualization* and *Mapping* options, 12/16 participants reported a preference for Lines, and 12/16 participants reported a preference for a Stable mapping. Participants also ranked 3 fingers first (7 participants), followed by 2 fingers second (7 participants), 1 finger third (5 participants), and 4 fingers last (9 participants) (Figure 4, right). Participants judged 4 fingers difficult to use because of (i) the awkward motor action to flex the pinky without flexing the ring finger, and (ii) the extra cognitive effort due to the additional color to remember.

To get a better understanding of these preferences, we also report the ratio between (i) the distance of the target and the virtual sphere, and (ii) the sphere's diameter (Figure 4, left). A ratio close to 0.5 would indicate that participants kept the target close to the volume boundaries (near the radius limit). We expected participants to have different strategies according to the mapping or the number of fingers used in each condition. For instance, with a Finger Mapping, users could position the desired target near the center of the virtual sphere, and hence preventing any change of color more likely near the sphere boundary. Interestingly, we did not find any particular pattern: with 1 finger (i.e., the standard Bubble Cursor), the ratio is 0.5 in case of success, and greater than 0.5 in case of error, indicating a snap-to impact. However, with 3 and 4 fingers, both Stable and Finger Mappings demonstrate the same average ratios.

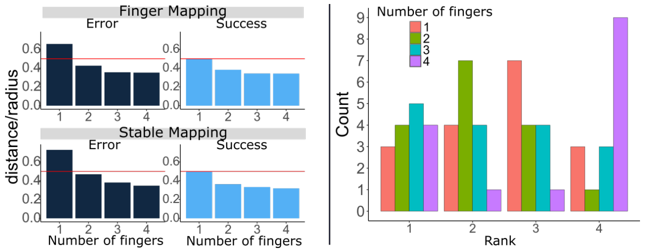

Figure 4: Left: Average ratio target distance over sphere diameter. Right: Preferences ranking for the *Number of fingers*.

Lastly, all participants commented on the occlusion with semi-transparent targets and colored halo. While they could see that the desired target was in the volume selection, they had difficulties perceiving its associated color, and hence the associated finger.

4 DISCUSSION AND FUTURE WORK

Based on our design experiment, we can discuss the final design we will use for future work. Surprisingly, we did not find any effect of *Visualization* or *Mapping* on user performances. However, participants express a strong preference for Lines with a Stable mapping. The design using 3 fingers is the fastest, most accurate, and also preferred by participants. However, we need to modify the halos so that targets further away from the user (i.e., the camera) have larger halos than targets in the front. We hypothesize that this mechanism can help user distinguish colors of occluded targets. Some errors happened because of the palm casting (Figure 3). We hence plan on providing a subtle cone emerging from the palm for users to perceive their aiming direction while moving the volume. We anticipate that this could help prevent the volume to suddenly switch cluster if not mindful of the palm orientation. Lastly, and more importantly, we plan on providing a 200 ms look-back mechanism: When detecting a finger flexion, the system selects the target associated with this finger 200 ms in the past. This look-back mechanism only impacts the trigger, i.e. without additional delay while interacting.

Our future work includes two additional experiments. First, we want to explore an adaptive density-aware behavior. Some areas or clusters in 3D environments might not require 3 fingers. Fingers could be included and excluded from the interaction process based on the current density. We first need to study if this adaptive behavior is well-perceived by users or if it brings a feeling of chaos and disorganization. Second, we want to compare MultiFingerBubble to close (i.e., fine-tuned 3D Bubble Cursor) and distant baselines: direct selection limited to clusters close to users, and RayCursor [6] requiring a specific device with 6 degrees-of-freedom tracking, a selection trigger, and a trackpad for the cursor. The goal is to compare these four selection techniques in different clusters with varying densities and distances to users.

5 CONCLUSION

We propose a new variant of the 3D Bubble Cursor for 3D target selection: MultiFingerBubble. With MultiFingerBubble, users aim at the cluster of targets of interest with their palm. Users can then control a semi-transparent sphere in the cluster with a 1:1 mapping between the sphere and users' hand positions. Instead of automatically pre-selecting the closest target like the 3D Bubble Cursor, MultiFingerBubble includes up to four targets where each target is then associated with a specific finger. Users can then validate the selection of the desired target by flexing the corresponding finger. A user experiment reveals that users prefer using MultiFingerBubble with lines, a stable mapping, and up to three fingers - which improves on the Bubble Cursor version using one finger only.

ACKNOWLEDGMENTS

This Project is financed by the Interreg Atlantic Area Program through the European Regional Development Fund (ERDF) from the European Commission.

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar. 2008. Improving 3D Selection in VEs through Expanding Targets and Forced Disocclusion. In *Smart Graphics (Lecture Notes in Computer Science)*, Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier, and Marc Christie (Eds.). Springer, Berlin, Heidelberg, 45–57. https://doi.org/10.1007/978-3-540-85412-8_5

[2] Ferran Argelaguet and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics* 37, 3 (May 2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[3] Felipe Bacim, Regis Kopper, and Doug A. Bowman. 2013. Design and Evaluation of 3D Selection Techniques Based on Progressive Refinement. *International Journal of Human-Computer Studies* 71, 7 (July 2013), 785–802. https://doi.org/10.1016/j.ijhcs.2013.03.003

[4] Gilles Bailly, Eric Lecolinet, and Yves Guiard. 2010. Finger-Count & Radial-Stroke Shortcuts: 2 Techniques for Augmenting Linear Menus on Multi-Touch Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 591–594. https://doi.org/10.1145/1753326.1753414

[5] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. 2011. Comparing Free Hand Menu Techniques for Distant Displays Using Linear, Marking and Finger-Count Menus. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II (INTERACT'11)*. Springer-Verlag, Berlin, Heidelberg, 248–262. https://doi.org/10.1007/978-3-642-23771-3_19

[6] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. 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, 1–12. https://doi.org/10.1145/3290605.3300331

[7] Anil Ufuk Batmaz, Xintian Sun, Dogu Taskiran, and Wolfgang Stuerzlinger. 2019. Hitting the Wall: Mid-Air Interaction for Eye-Hand Coordination. In *25th ACM Symposium on Virtual Reality Software and Technology (VRST '19)*. Association for Computing Machinery, New York, NY, USA, 1–5. https://doi.org/10.1145/3359996.3364249

[8] Joanna Bergström, Tor-Salve Dalsgaard, Jason Alexander, and Kasper Hornbæk. 2021. How to Evaluate Object Selection and Manipulation in VR? Guidelines from 20 Years of Studies. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–20. https://doi.org/10.1145/3411764.3445193

[9] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. *3D User Interfaces: Theory and Practice.* Addison Wesley Longman Publishing Co., Inc., USA.

[10] Doug A. Bowman and Chadwick A. Wingrave. 2001. Design and Evaluation of Menu Systems for Immersive Virtual Environments. In *Proceedings IEEE Virtual Reality 2001.* 149–156. https://doi.org/10.1109/VR.2001.913781

[11] Jeffrey Cashion, Chadwick Wingrave, and Joseph J Laviola Jr. 2012. Dense and Dynamic 3D Selection for Game-Based Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (April 2012), 634–642. https://doi.org/10.1109/TVCG.2012.40

[12] Gerwin de Haan, Michal Koutek, and Frits H. Post. 2005. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Proceedings of the 11th Eurographics Conference on Virtual Environments (EGVE'05)*. Eurographics Association, Goslar, DEU, 201–209.

[13] P M Fitts. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of experimental psychology* 47 (Sept. 1954), 381–391. https://doi.org/10.1037/h0055392

[14] Tovi Grossman and Ravin Balakrishnan. 2005. 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.* Association for Computing Machinery, New York, NY, USA, 281–290. https://doi.org/10.1145/1054972.1055012

[15] Regis Kopper, Felipe Bacim, and Doug A. Bowman. 2011. Rapid and Accurate 3D Selection by Progressive Refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI).* IEEE, Singapore, Singapore, 67–74. https://doi.org/10.1109/3DUI.2011.5759219

[16] G. Kurtenbach, T. P. Moran, and W. Buxton. 1994. Contextual Animation of Gestural Commands. *Computer Graphics Forum* 13, 5 (1994), 305–314. https://doi.org/10.1111/1467-8659.1350305

[17] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655

[18] Huiyu Li and Linwei Fan. 2019. A Flexible Technique to Select Objects via Convolutional Neural Network in VR Space. *Science China Information Sciences* 63, 1 (Dec. 2019), 112101. https://doi.org/10.1007/s11432-019-1517-3

[19] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR).* 35–43. https://doi.org/10.1109/VR46266.2020.00021

[20] Peter Mohr, Markus Tatzgern, Tobias Langlotz, Andreas Lang, Dieter Schmalstieg, and Denis Kalkofen. 2019. TrackCap: Enabling Smartphones for 3D Interaction on Mobile Head-Mounted Displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300815

[21] Sergio Moya, Sergi Grau, and Dani Tost. 2013. The Wise Cursor: Assisted Selection in 3D Serious Games. *The Visual Computer* 29, 6 (June 2013), 795–804. https://doi.org/10.1007/s00371-013-0831-3

[22] Soonchan Park, Seokyeol Kim, and Jinah Park. 2012. 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).* Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/2350046.2350060

[23] Jordan Pollock and Robert J. Teather. 2020. CountMarks: Multi-finger Marking Menus for Mobile Interaction with Head-Mounted Displays. In *HCI International 2020 - Late Breaking Papers: Multimodality and Intelligence (Lecture Notes in Computer Science)*, Constantine Stephanidis, Masaaki Kurosu, Helmut Degen, and Lauren Reinerman-Jones (Eds.). Springer International Publishing, Cham, 241–260. https://doi.org/10.1007/978-3-030-60117-1_18

[24] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. 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).* Association for Computing Machinery, New York, NY, USA, 79–80. https://doi.org/10.1145/237091.237102

[25] Yuan Yuan Qian and Robert J. Teather. 2017. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17).* Association for Computing Machinery, New York, NY, USA, 91–98. https://doi.org/10.1145/3131277.3132182

[26] Gang Ren and Eamonn O'Neill. 2013. 3D Selection with Freehand Gesture. *Computers & Graphics* 37, 3 (May 2013), 101–120. https://doi.org/10.1016/j.cag.2012.12.006

[27] Shana Smith and Emily Ericson. 2009. Using Immersive Game-Based Virtual Reality to Teach Fire-Safety Skills to Children. *Virtual Reality* 13, 2 (June 2009), 87–99. https://doi.org/10.1007/s10055-009-0113-6

[28] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2006. Object Selection in Virtual Environments Using an Improved Virtual Pointer Metaphor. In *Computer Vision and Graphics: International Conference, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings*, K. Wojciechowski, B. Smolka, H. Palus, R.S. Kozera, W. Skarbek, and L. Noakes (Eds.). Springer Netherlands, Dordrecht, 320–326. https://doi.org/10.1007/1-4020-4179-9_46

[29] Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces.* https://doi.org/10.1109/3DUI.2007.340783

[30] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Fully-Occluded Target Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (Dec. 2020), 3402–3413. https://doi.org/10.1109/TVCG.2020.3023606