Design and Evaluation of Visualization Techniques of Off-Screen and Occluded Targets in Virtual Reality Environments

Difeng Yu, Hai-Ning Liang, Kaixuan Fan, Heng Zhang, Charles Fleming, Konstantinos Papangelis

Abstract—This research explores the design and evaluation of visualization techniques of targets that reside outside of users' view or are occluded by other elements within a virtual reality environment (VE). We first compare four techniques (3DWedge, 3DArrow, 3DMinimap, and Radar) that can provide direction and distance information of targets. To give structure to their evaluation, we also develop a framework of four tasks (one for direction and three for distance) and their assessment criteria. The results show that 3DWedge is the best-performing and most usable technique. However, all techniques, including 3DWedge, have poor performance in dense scenarios with a large number of targets. To improve their support in dense scenarios, a fifth technique, 3DWedge+, is developed by using 3DWedge as its foundation and including the visual elements of the other three techniques that are useful. A second study is conducted to evaluate the performance of 3DWedge+ in relation to the other techniques. The results show that both 3DWedge and 3DWedge+ are significantly better in distinguishing user-to-target distance and that 3DWedge+ is particularly suitable for dense scenarios. Based on these results, we provide a set of recommendations for the design of visualization techniques of off-screen and occluded targets in 3D VE.

Index Terms—Evaluation; head-mounted displays; occluded targets; off-screen targets; virtual reality; visualization techniques.

1 Introduction

Locating targets of interest in a 3D environment often becomes difficult when they reside outside the user's view or are occluded by other objects in the environment [24, 25]. Because these objects are not directly visible to users, it will not be possible for them to perceive how far and in which direction the objects are located in the environment. As such, even the simple, yet quite common, task of looking for them cannot take place. Fig. 1 presents an example scenario. A user wants to find all nearby friends in a 3D virtual reality environment (VE) and get a sense of their relative locations so that she can then plan an efficient route to reach them. This task will be very challenging for the user as she can only see a couple of her friends, while the others are occluded by buildings or reside outside of her view (Fig. 1a). This scenario can often be seen in VR social meeting environments, like VRChat (vrchat.net), where users who are represented as avatars meet each and explore places together in 3D virtual worlds like restaurants, discos, buildings, movie theaters, etc.

Current common solutions for visualizing multiple objects in an environment include 2D maps [1, 6] and 3D Arrow Clusters [3]. However, 2D maps are not able to differentiate targets with different heights [4]. Height information can play an important role for targets located inside a multi-layer building, such as a large shopping mall, which is quite common in VE. Furthermore, current 3D Arrow techniques need to use additional text information to show distance. Text is not easy to read and make comparisons with. In addition, a large amount of text will clutter the visualization and further increase the difficulty of reading it [28, 29, 30].

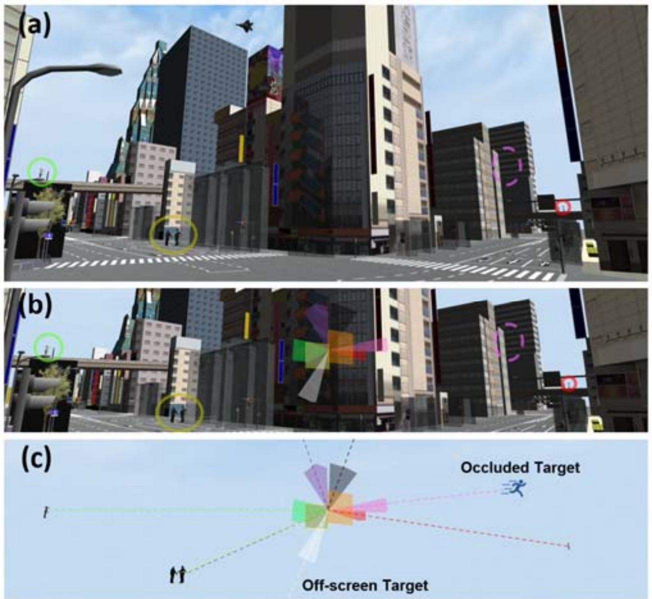


Fig. 1. A common scenario in a multi-user 3D virtual reality world: To locate all nearby targets (e.g., avatars of people or friends). (a) From a user's point of view, targets of interest may be located off-screen, be occluded by other objects, or placed inside buildings. (b) An example of embedding a visualization (in this case 3DWedge) into the user VR interface to convey targets' information. (c) 3DWedge is designed to provide accurate direction and distance information of targets through the pyramid and other visual elements.

- *Difeng Yu, Hai-Ning Liang, Kaixuan Fan, Heng Zhang, Charles Fleming, and Konstantinos Papangelis are with the Department of Computer Science and Software Engineering at Xi'an Jiaotong-Liverpool University, Suzhou, China.*
- *Emails: Difeng.Yu14@student.xjtlu.edu.cn; HaiNing.Liang@xjtlu.edu.cn; Kaixuan.Fan16@student.xjtlu.edu.cn; Heng.Zhang14@student.xjtlu.edu.cn; Charles.Fleming@xjtlu.edu.cn; K.papangelis@xjtlu.edu.cn*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

The goal of this research is to develop techniques to assist users in locating multiple targets in a 3D VE quickly and accurately. To achieve this goal, we first examined research on the visualization of objects in other domains and then developed four initial visualization techniques (3DWedge, 3DArrow, 3DMinimap, and Radar) for a head-mounted display (HMD) based 3D VE. We conducted a study to evaluate the techniques using one direction task and three distance tasks. Our results indicate that 3DWedge (Fig. 1b-c) performs especially well among the four and is well-liked by participants. On the other hand, the first study also shows that all techniques, including 3DWedge, suffered in dense environments with many targets.

To explore the possibility of a technique that could perform well even in dense cases, we decided to devise a fifth technique 3DWedge+, by using 3DWedge as its foundation and adding the positive visual features of the other three techniques. A second study was conducted to compare 3DWedge+ with the three best performing techniques from the first study. The results show that 3DWedge+ significantly increase its accuracy in high-density configurations. As different types of visualizations, which use different visual elements, were explored in these two studies, we were able to identify the visual elements that were useful. As such, we have been able to distill recommendations for designing visualization techniques of targets in 3D VE, especially those that are occluded or reside outside of the field-of-view of HMD.

In the remainder of this paper, we first review related work about the visualization of targets in 2D/3D environments and visualization tasks. We then describe in detail the four candidate techniques and the experimental framework of our work including the VR environment setting, tasks, and the evaluation criteria. After, we present the two user studies and their results. At the end of the paper, we provide recommendations extracted from the results for the design of the visualizations of targets in 3D VE.

2 Related Work

In the following section, we first describe relevant studies on off-screen target visualization in 2D environments. We then present some existing techniques for displaying off-screen targets in 3D environments. In addition, related work about the design of the experimental tasks to assess the performance of the techniques is reviewed and used for framing our two experiments.

2.1 Visualizations of targets in 2D environments

Researchers have investigated and proposed off-screen location visualization techniques in mobile devices [32]. City Light [8], a space-efficient fisheye technique, describes unseen targets in all directions from a focused view. Halo [1] uses rings to surround off-screen objects to help the user to infer the off-screen location of the targets. Wedge [9] represents an improvement over Halo because, by using acute isosceles triangles instead of arcs to convey location information, it minimizes visual overlaps and clutter. EdgeRadar [10, 11] implements another fisheye-based view to track off-screen targets and has been found to elicit a slightly higher preference from users over Halo. Canyon [12], which employs the paper-folding metaphor to provide context around target locations, is shown to be more accurate than Wedge across complex tasks. Both Canyon and EdgeRadar do not seem to be easily transferable to 3D HMD VE. Also, their designs do not lend themselves easily to represent relative location information.

Scaled-Arrows [2] and Mini-map [14, 20] are also suggested to perform well in 2D off-screen visualization tasks. A recent approach called Personalized Compass [13], which combines a multi-needle compass with an abstract overview map, has been shown to complement Wedge in inferring the targets' location and direction information. Although these techniques cannot be applied to the 3D scenario directly since the height information of targets is unavailable, they have the potential to aid the framing of techniques for 3D HMD environments. They have inspired the techniques that are developed and explored in this research.

2.2 Visualizations of targets in 3D environments

The 3D arrow approach is designed for navigation tasks and provides cues to help identify targets of interest [3]. This technique involves 3D arrows pointing to specific targets, with supplementary text to show distance information. However, text labels are often difficult to read and, when there are many of them, they will occlude each other. Low readability is not desirable as it adds an extra visual and cognitive burden to users and decreases performance [28, 29, 30]. Schinke et al. [15] have proposed scaled-down 3D arrow cues for augmented reality (AR) applications. Their technique scales the length of the arrow according to the distance between the viewer and objects of interest, and it is shown to be more efficient than traditional 2D map-like techniques for some tasks. Nevertheless, because their technique places the center of each arrow tip on the circle boundary, the arrows have to be located on the same plane and as such cannot show the height information of the targets.

Aroundplot [4] is another technique that provides multiple location cues for off-screen objects to help guide the user towards these targets in a 3D environment. It maps the objects in 3D to a 2D orthogonal fisheye view [23] and is shown to be more accurate than 3D Arrow in determining target direction when the number of objects is large. Two other techniques, Parafrustum [34] and Attention Funnel [36], have also been proposed for AR applications to help users find specific targets in a 3D environment [38]. Another visualization technique, SidebARs [16], implements two sidebars to allow users to see distance and direction information quickly. The main disadvantage of all the above techniques is that they are not able to show distance information without the use of text. In addition, users have to turn their head frequently to get accurate direction information for each target one by one.

More recent approaches have tried to adapt work from 2D visualizations into 3D environments [33]. For example, 3D Halo Circle [18] and Halo3D [17] can display 3D direction and distance information of off-screen targets and are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

adaptations of 2D Halo for mobile phones. However, similar to the 2D version, 3D Halo Circle still has visual clutter and overlap issues [17]. 3DHalo is not effective because it does not help users distinguish if the targets are in front of them or behind them.

3D map techniques come from the early work of 'Worlds in Miniature' (WIM) [7] which augments an immersive display with a hand-held miniature copy of the virtual environment. Chittaro et al. [19] adapted 3D maps as navigation aids for virtual multi-floor buildings. They use a small-scale miniature map to show the whole building but with much less detail. Because their results seem positive, it is worth exploring how well 3D map techniques might be adapted to help visualize targets in HMD.

Most current techniques have focused on giving users direction information. In 3D VE, knowing the distance of the targets from the user's location is also important, but little research has been done on this aspect. In addition, as is the case of Halo and 3D Halo Circle, visual clutter should be minimized to ensure that users can have unobstructed views of the 3D VE. Minimizing visual clutter is quite important given that the current HMDs have a small field-of-view [35].

To the best of our knowledge, there does not seem to be a visualization of targets for HMD VE yet. To narrow this gap, we have initially designed four techniques. Each technique has a different foundation, base representation and is designed to help users assess the distance of targets and the direction of the targets' location. The results of an experiment with these techniques led to the development of a fifth technique that brings together the useful visual features of all four initial techniques.

2.3 Target visualization experimental task design

The performance of the visualization techniques is usually measured by how fast and accurate users can perform specific tasks. Our review of the literature has helped us identify several tasks that are mainly used in 2D scenarios. The experiment with Halo [1] consists of four experimental tasks (“*Locate*”, “*Closest*”, “*Traverse*”, and “*Avoid*”) which are derived from real problem-solving situations. These tasks require the participants to locate expected locations of off-screen targets, select the closest target to the user, traverse through all targets using the shortest path, and select the target that is farthest away from some location (e.g., jam-packed roads). Burigat et al. [2, 14] have introduced another two tasks: “*Estimate*” and “*Order*”. These two tasks ask users to determine the closest pair of targets and order the targets in increasing distance from the center of the map. These two tasks are the result of aggregating smaller subtasks that deals with spatial awareness and reasoning. The above six tasks are mainly about determining the distance of objects. In addition to distance tasks, a recent work [13] uses the “*Orientation*” task, which asks users to determine the direction of off-screen targets.

There has also been some research on designing tasks for 3D environments. Schinke et al. [15], for example, have used a task which requires the user to turn to the direction of the targets and then either read the name of the targets or memorize them. Additionally, Jo et al. [4] have proposed two search tasks: normal search and highlighted search. These two tasks mainly focus on finding the direction of the targets' location. Unlike 2D scenarios, the emphasis is on identifying the direction of objects. However, like 2D scenarios, knowing the distance of objects is also important so that users can determine if objects are close to or far from them and plan efficient paths to reach all the targets of interest.

In summary, tasks can be grouped into direction (or orientation) and distance. Our review also has helped us determine three distance tasks that deal with both target-to-target and user-to-target distances and one direction task. These four tasks and their assessment criteria are described in detail in Section 4.3 below.

3 The Visualization Techniques

As a starting point, we examined techniques from 2D and 3D environments to guide our design of techniques for 3D HMD VE. We required the techniques to show both distance and direction information of the targets. Moreover, they should take up as little screen space as possible. To determine which techniques are suitable, we first developed a series of prototypes and ran some pilot tests with them. In the end, four final candidate techniques were selected, namely 3DWedge, 3DArrow, 3DMinimap, and Radar. Before introducing the techniques, we first present four general common visual elements a target visualization technique could be composed of:

- **Target representation**. This stands for the elements that show the target(s) in the visualization.
- **User-target indicator**. This is the visual element that connects the target representation(s) to the user's reference point in the visualization.
- **Direction indication**. This represents the direction that the user is looking towards in the visualization.
- **Distance indication**. This refers to how the distance information of the target(s) is represented.

We next present the four visualization techniques and in Table 1 further summarize how the components of these techniques can be categorized into the above basic visual elements.

3.1 3DWedge

3DWedge is shaped like a pyramid with a square base (Fig. 2b). The height of the pyramid is linearly scaled according to the distance between the viewer and the target. The

Algorithm 1. Calculating the base length $F(i)$ of the 3DWedge corresponding to the i[th] closest target

n ← pre-defined smallest base length
m ← pre-defined largest base length
p ← the number of targets in the scene
if $p > 1$ **then**
  $F(i) = n + \frac{m-n}{p-1} \times (i-1)$
else
  $F(i) = n$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

length of the sides of its square base, which we call base length, is non-linearly scaled based on the target's distance according to Algorithm 1. The equation in the algorithm (line 5) creates different "levels" for the base length of several 3DWedges when there are multiple targets in the VE. By comparing these "levels", the users will be able to differentiate distances of several targets. Our pilot study shows that the technique works well even in situations where several targets are located within proximity to each other (Fig. 2c-d). By comparing these levels of the bases, users can find it easy to determine the distances between the targets and her location accurately.

To minimize visual obstruction, we place the 3DWedge at an angle θ below the main direct view of the user. However, this can make it difficult for the user to accurately determine the direction that the 3DWedge components are pointing towards from the user's view (Fig. 2e). To solve this issue, the 3DWedge cluster is rotated with the angle θ to compensate for the shift (Fig. 2f). This allows the user to determine the direction of the targets by only looking at the 3DWedge (Fig. 2g). We also set the 3DWedge to be transparent and remove all lines to reduce visual obstruction and clutter further.

Fig. 2a presents a scenario with 3DWedge. Several 3DWedges are anchored to a single reference point to create a cluster. The whole cluster shows the distance and direction of the targets located in the VE. This visualization and the other three techniques update dynamically as the user looks or moves around the scene.

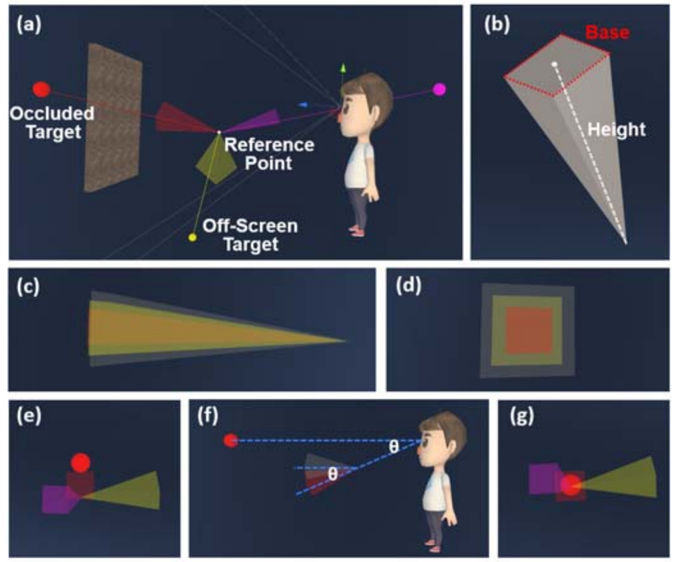


Fig. 2. (a) A user scenario with 3DWedge. (b) The square base and the height of 3DWedge. The left view (c) and the front view (d) of a 3DWedge cluster when three targets with similar distances and directions exist in the scene. (e) Without direction adjustment, the user will have difficulty sensing the accurate direction of the target when 3DWedge is placed below the main view of the user (f) the direction adjustment for 3DWedge (from the white direction to the red direction); (g) the user can estimate the precise direction of the target after adjusting the direction.

3.2 3DArrow

3DArrow, which contains an arrow tip and a stick (see Fig. 3a), is motivated by previous work in 2D [2] and 3D [3, 15]. A 3DArrow cluster is a collection of 3DArrows tied to a central vertex (which we call a reference point). Each 3DArrow points in the direction of a target and the length of the stick is linearly scaled as a function of the distance between the user and the target.

3.3 3DMinimap

Inspired by related work in 2D [6, 14, 20] and 3D [7, 19] scenarios, our 3DMinimap is composed of three layers of concentric spheres, a user model, the simulation of the users' field-of-view and scaled down targets that are shown as small spheres. The three levels of spheres help users gauge the targets' distance. The yellow pyramid (Fig. 3b) is used to show the user's field of view (FoV). This design is inspired by a technique in 2D that has been shown to be useful in facilitating navigation tasks [6]. The red line represents the center view of the viewer, which gives users a visual element to help determine their viewing direction. A user model is placed in the middle of the sphere to represent the user's location, and the small circles are the targets that are in the environment. When the user needs to move their head to look around, the FoV, the central view, and the spheres with the targets will move in relation to head movement in real time. This will allow users to visualize the direction of the targets with ease [6].

3.4 Radar

Radar is based on 2D radar visualization techniques [4]. Targets are represented as small squares. Text is added to convey the horizontal distance and height information of the targets (Fig. 3c). The visualization also has a thick highlighted line coming out from the middle to represent the direction that the user is looking towards. This visual element, which we call line view, is added to simulate the user's view when the user rotates their head horizontally. During our early tests with the technique, we found that in dense configurations, text from targets would often overlap one another. To minimize this issue, we added a collision checking algorithm so that the text of one target would not occlude the text of other targets.

TABLE 1
VISUALIZATION TECHNIQUES AND CORRESPONDING VISUAL ELEMENTS

	Target Representation	User-target Indicator	Distance Indicator	Direction Indicator
3DWedge	Square base	Pyramid	Pyramid height, base length	Middle of the square base
3DArrow	Arrow tip	Stick	Stick length	Arrow tip
3DMinimap	Small sphere	-	Concentric spheres	Center view, FoV
Radar	Small square	-	Text, concentric circles	Line view

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

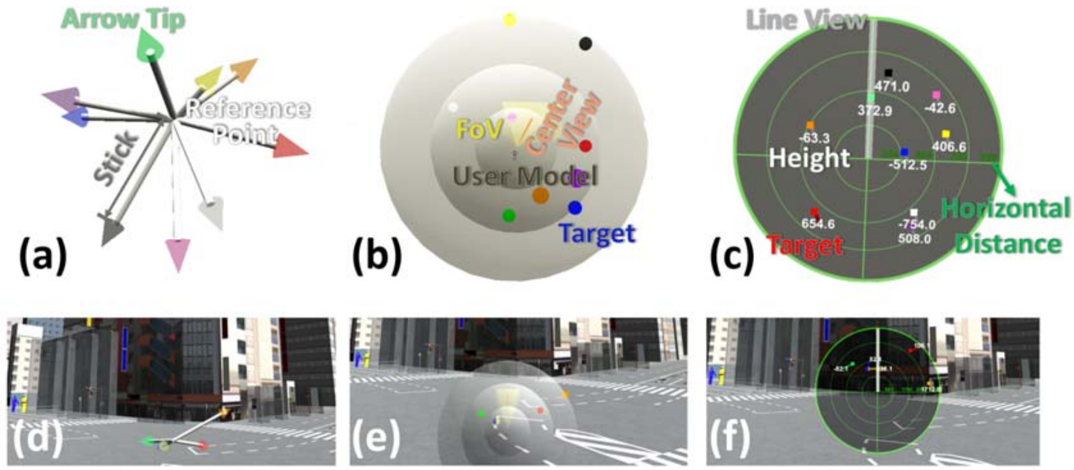


Fig. 3. Apart from 3DWedge, we also developed three other techniques; each color implies a distinct target: (a) 3DArrow: contains a stick and an arrow tip. All 3DArrows start from the reference point; (b) 3DMinimap: contains a user model in the center, and the targets are distributed inside the sphere. The direction that the user is looking towards is represented using FoV and Center View; and (c) Radar: texts are provided to represent both the horizontal distance and height information of targets. The direction that the user is looking towards is simulated using a line. (d-f) How the visualizations are shown in the HMD. All the figures are obtained from a 3D view and because of this, objects further away look smaller.

4 Experiment Design Framework

In this section, we describe the design framework of our experiments. We first present the experimental environment and the apparatus. We later describe the four tasks and their evaluation criteria.

4.1 Experimental Environment

The VE was built as an urban area consisting of roads and many buildings and was similar to the virtual worlds in VRChat and other virtual social environments. We randomly generated the targets in the scene and made sure that they were all located outside of the visual range of the user, so that they were either outside of the user's field-of-view or occluded by other objects.

Prior research had shown that selecting targets with different sizes could lead to a different index of difficulties [21]. To minimize any influence due to selection, we had our participants use the “selection box” to select the targets. Fig. 4 shows the selection boxes for two density configurations that either had 4 or 9 circles of different colors. Each circle corresponds to the object of the same color in the environment.

4.2 Apparatus and Materials

The experiment was conducted using a PC that had an i7 CPU and an NVIDIA 1080 GPU. The program was developed using C#.NET and was run within the Unity3D platform.

We used the Oculus RIFT CV1, an HMD virtual reality device that could completely immerse the user into the 3D virtual world and allow the users to look in any direction by simply rotating their head. The wireless Oculus Touch hand controller was used to make the selection on the selection box. The index trigger of the Touch was used to confirm the selection, and another button was used to proceed from one part of the study to the next. The ray-casting technique [31] was used to show which targets in the selection box would be selected when the trigger was pressed (see Fig. 4c).

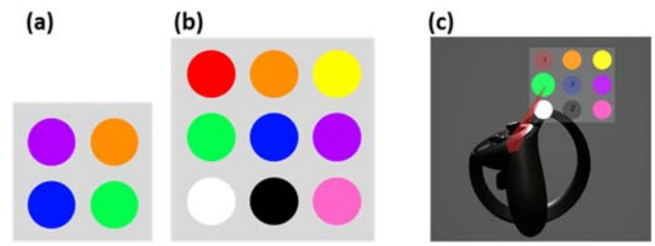


Fig. 4. (a-b) The selection box used for selection confirmation. (c) The selection button is pressed.

4.3 Tasks and Evaluation Criteria

The experimental tasks were conceptualized from our review of target selection literature [2, 13]. Our main objective was to evaluate the efficiency of the techniques in estimating the distance and direction of the targets in relation to the users' location. In addition to the tasks we also identified their assessment criteria. Fig. 5 shows the evaluation framework of our experiment.

4.3.1 The “Closest to User” Task

Because several targets would be randomly generated and placed in the 3D environment within a certain range from

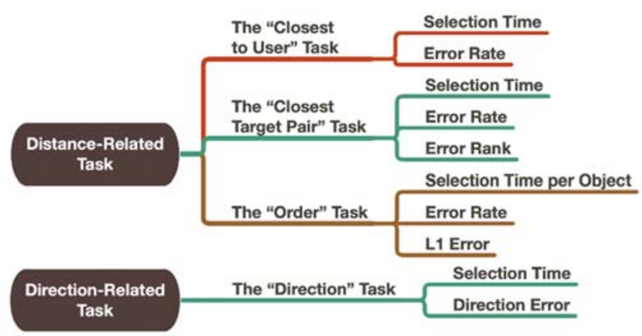


Fig. 5. The evaluation framework with four tasks and their corresponding assessment criteria.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

the users, this task would require participants to select the target closest to their location. For this task, we recorded the selection time, which indicated how much time the participants needed to confirm the selection. Errors were also recorded when they did not select the right target.

4.3.2 The "Closest Target Pair" Task

For this task, participants were asked to select the pair of targets which were closest to each other. To complete the task, they should estimate the closest pair of targets and select these two targets in the selection box. Selection time and the number of errors were also recorded. Moreover, we also introduced error ranking for evaluating the relative magnitude of the errors of the techniques. To calculate the error rank, we followed these three steps:

1. Order all the distances between the different target pairs in ascending order as a list L which would thus have $n \times (n-1)/2$ elements, where n is the number of targets in the environment.
2. Find the location of the selected pair in the list L and save its rank in the list as p.
3. Error Rank $= p - 1$.

For instance, if one participant selected the third closest pair, the error rank would be 2. By using this approach, we could assign different weightings based on how correct (or incorrect) participants' selections were. For example, selecting the pair of objects with the longest distance between each other would receive a high penalty while selecting the second shortest distance pair would have a smaller contribution to the final error.

4.3.3 The "Order" Task

In this task, participants were required to order all the targets in increasing distance from their location. This task was designed to test if the techniques would allow users to distinguish distances accurately, even if two distances were small. To complete this task, users should select all the circles on the selection box based on their ascending distance order.

Selection time for each object, which corresponded to how long a participant spent to make the selection, was calculated by dividing the total time by the number of targets in the trial. An error would be recorded when the participant selected the objects in the wrong order. Furthermore, we computed the *L1 error* by using the following method. Suppose the vector $\boldsymbol{p} = (p_1, p_2, \dots, p_n)$ was the participant's selected order and the vector $\boldsymbol{q} = (q_1, q_2, \dots, q_n)$ was the correct order. The L1 error would be:

$$\text{L1error}(\boldsymbol{p}, \boldsymbol{q}) = ||\boldsymbol{p} - \boldsymbol{q}||_1 = \sum_1^n |p_i - q_i| \quad (1)$$

Similar to rank error, we could assign different penalties based on the relative magnitude of the error. For example, given the correct order was $\boldsymbol{q} = (1, 2, 3)$, but the selected order was $\boldsymbol{p_1} = (3, 2, 1)$, we would have the $\text{L1error}(\boldsymbol{p_1}, \boldsymbol{q}) = 4$. If, on the other hand, the selected order was $\boldsymbol{p_2} = (1, 3, 2)$ we would have the $\text{L1error}(\boldsymbol{p_2}, \boldsymbol{q}) = 2$ instead. In this example, $\boldsymbol{p_1}$ would receive a higher penalty than $\boldsymbol{p_2}$ because in the former case, the participant's mistake was ranked as more significant. Moreover, because selecting all the objects in the right order was very difficult in dense configurations, using this approach would allow measuring the relative performance of the four techniques based on *how* correct the ordering of the targets was.

4.3.4. The "Direction" Task

This task was to find the direction of one given target in the 3D environment. The location of the selected target and distractor targets were randomly generated in the scene. To complete the task, the participants would need to use the reticle to point in the direction of the highlighted target and confirm the selection.

Both selection time and direction error were recorded. Direction error was calculated as the angle between the estimated direction vector and the actual direction vector of the object from the participant's point of view. The minimum direction error would be zero, which means the two vectors were parallel to each other and in the same direction, while the largest direction error will be 180 which means the two vectors were in the opposite direction. Moreover, we also recorded the *appearance angle*, which represented the horizontal angle between the participant's viewing vector when the target appeared and the actual direction vector, to analyze whether it would affect the selection time. To do this, we divided the space into three areas (Front, Left/Right, and Back). As the field-of-view of the Oculus RIFT was 96 degrees, the Front Area is set to be the angle from 0–48 degrees, the Left/Right Area from 48–132 degrees, and the Back Area from 134–180 degrees.

5 Study 1: Evaluating the Four Techniques

In this first study, we compared 3DWedge, 3DArrow, 3DMinimap, and Radar across the four tasks (Closest to User, Closest Target Pair, Order, and Direction) with two levels of density configurations (four and nine targets). The goal of this study was to evaluate the four techniques regarding selection time and accuracy. In addition, we wanted to know what visual features or elements of the techniques would be useful.

To make fair comparisons by considering the speed-accuracy tradeoff, we defined here that one technique (A) performed better than another technique (B) if and only if one of the three following conditions was satisfied:

1. A required significantly less selection time and yielded significantly higher accuracy than B; or
2. A required significantly less selection time than B, and there was no significant difference in accuracy between the two techniques; or
3. A yielded significantly higher accuracy than B, and there was no significant difference in selection time between the two techniques.

5.1 Hypotheses

For this experiment, we had these four hypotheses:

H.1. *For the "Closest to User" task, 3DWedge, 3DArrow, and 3DMinimap would perform better than Radar*. We expected that the distances between the user and targets could be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

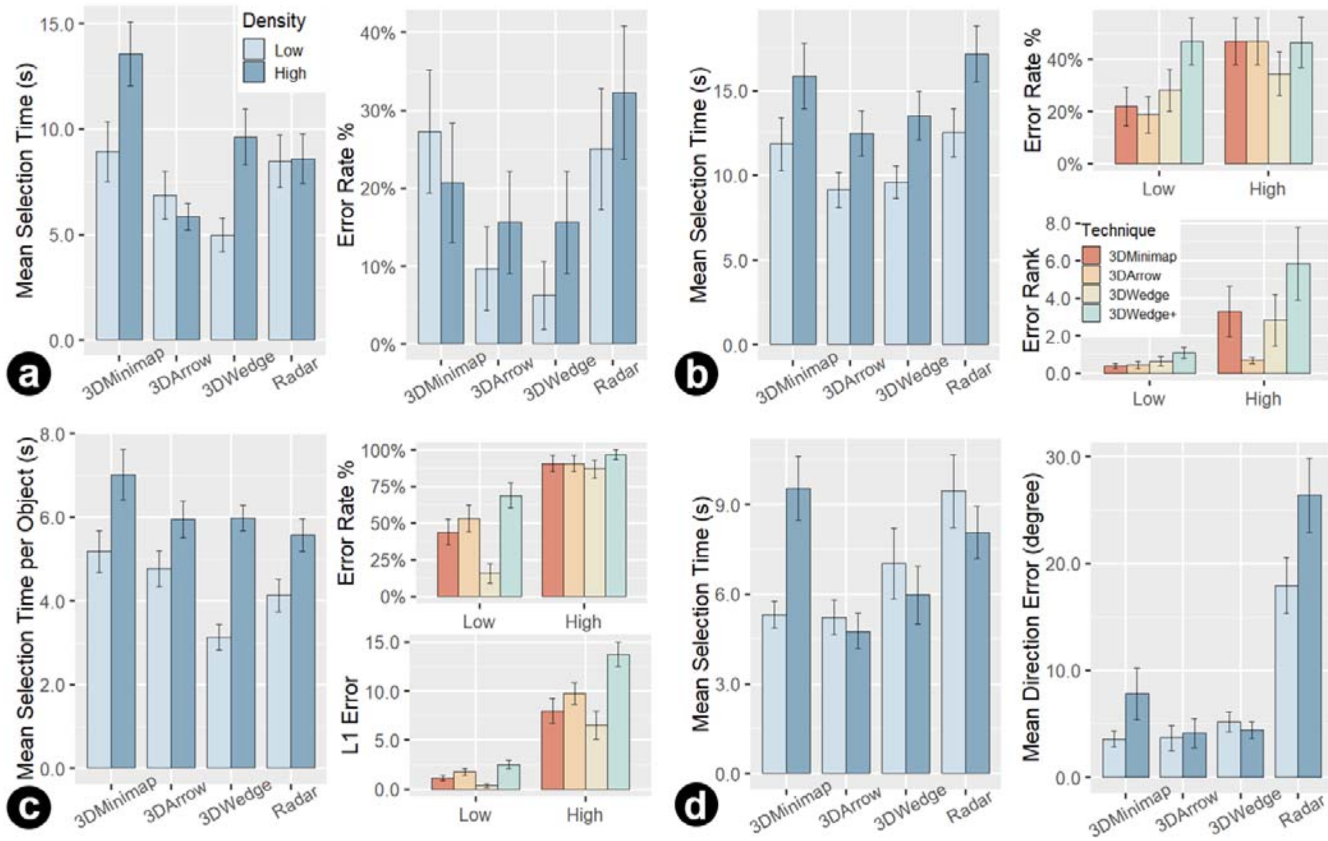


Fig. 6. (a) The “Closest to User” Task mean selection time and error rate; (b) the “Closest Target Pair” Task mean selection time, error rate and error rank; (c) the “Order” Task mean selection time per object, error rate and mean L1 Error; and (d) the “Direction” Task mean selection time and mean direction error. Error bars all indicate standard error.

compared easily through the height of a 3DWedge and the stick length of a 3DArrow. The three spheres of 3DMinimap would also help with distance comparisons. However, it would not be easy for participants to assess the distance using the text in Radar.

H.2. *For the “Closest Target Pair” task, 3DArrow and 3DMinimap would perform better than 3DWedge and Radar*. The arrow tips of 3DArrow and circles representing targets in the concentric circles of 3DMinimap would allow more accurate comparisons of the distances between targets. The square base of 3DWedges and the projected target representations of Radar could lead possibly to less accurate comparisons.

H.3. *For the “Order” task, 3DWedge would perform better than all the other techniques*. By using non-linear scaling of the targets, participants could be able to distinguish via the length of the bases of each wedge the distance between their location and the targets with more ease and accuracy than with the other three techniques that followed a linear scaling.

H.4. *For the “Direction” task, 3DWedge, 3DArrow, and 3DMinimap would perform better than Radar*. The arrow tip of 3DArrow and the center view of the 3DMinimap would allow participants to point accurately towards the direction of the targets. They would also be able to sense the pointing direction of a wedge because of its square base (e.g., the flatter it is, the smaller the difference of the angle). Radar would not show the heights of the targets intuitively, and this would make the task difficult.

5.2 Participants, Experiment Design and Procedure

Sixteen participants (4 female, 12 male) between the ages of 19-23 (M=20) were recruited from a local university campus. Data from the pre-experiment questionnaire indicated that 8 participants had some basic experience with VR before. A pre-experiment color test showed that all of them could distinguish the colors we used in the selection box. They all had normal or corrected-to-normal vision.

The experiment lasted approximately 45 minutes for each participant. Before the trials started, they were given time to become familiar with the VR environment and the selection mechanisms. They were then asked to practice using the four techniques and tasks in a practice environment. They were allowed to ask any questions during this period. After this initial stage, they proceeded to carry out the four tasks. We instructed the participants to complete each task as quickly and accurately as possible. A trial ended when a participant confirmed the selection. A short sound would be played to indicate that the trial was complete. A rest scene would pop up after the completion of a task. After the experiment, participants were asked to complete a post-experiment questionnaire to collect their preference about the different tasks and comments about the techniques.

The study used a 4×2 within-subjects design with two factors: Technique (3DMinimap, 3DArrow, 3DWedge, or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

Radar) and Density (4 or 9 Targets). We fully counterbalanced the order of tasks and techniques using the Latin Square approach. The sequence of the two different dense environments was generated in random order. We repeated each trial two times in this experiment. Thus, for each task, we gathered 4 (technique) × 2 (density) × 2 (block) × 16 (participant) = 256 timed trials.

5.3 Experimental Results

We organized the results based on the hypotheses formulated earlier. Fig. 6 summarizes the performance of the four techniques based on the four tasks. We observed that participants sometimes took an exceptionally long time to compare minor differences. Therefore, for each task, we removed these outliers, which were those that had selection times of more than three standard deviations from the mean (for the Order task, it was the selection time per object). Because of this, the "Closest to User" and the "Closest Target Pair" tasks both dropped 4 trials (~1.6%), leaving still 252 trials. The "Order" and the "Direction" tasks both dropped 7 trials (~2.7%), leaving 249 trials.

For **H.1**, a 4 × 2 (Technique × Density) ANOVA was performed on selection time and error rate with the "Closest to User" task. The results showed that there were significant effects of Techniques on both selection time ($F_{3, 45}$ = 6.372, p<.001, η^2_P = .073) and error rate ($F_{3, 45}$ = 3.085, p<.05, η^2_P = .037). Post-hoc Tukey tests indicated that 3DArrow (p<.001) and 3DWedge (p<.01) were significantly faster than 3DMinimap but found no significant difference among the techniques concerning the error rate.

For **H.2**, a 4 × 2 (Technique × Density) ANOVA was performed on selection time, error rate, and error rank with the "Closest Target Pair" task. Results showed a significant main effect of Techniques on selection time ($F_{3, 45}$ = 3.431, p<.05, η^2_P = .040) and error rank ($F_{3, 45}$ = 3.155, p<.05, η^2_P = .037), but not on error rate ($F_{3, 45}$ = 1.339, p>.1, η^2_P = .016). Post hoc Tukey tests showed that 3DArrow was significantly faster than Radar (p<.05) and that error rank of 3DArrow was significantly lower than Radar (p<.05).

For **H.3**, a 4 × 2 (Technique × Density) ANOVA was performed on selection time, error rate, and L1 error with the "Order" task. The results showed that Technique had significant effect on selection time per object ($F_{3, 45}$ = 4.888, p<.005, η^2_P = .057), error rate ($F_{3, 45}$ = 6.897, p<.001, η^2_P = .079), and L1 error ($F_{3, 45}$ = 9.529, p<.001, η^2_P = .106). Post-hoc analysis revealed that 3DWedge (p<0.01) and Radar (p<.05) were significantly faster than 3DMinimap. Moreover, 3DWedge led to much lower errors than 3DArrow (p<.05) and Radar (p<.001). However, in the high-density configurations with 9 targets, no significant differences were found among the four techniques ($F_{3, 45}$ = 0.588, p>.5, η^2_P = .015).

For **H.4**, a 4 × 2 × 3 (Technique × Density × Target Appearance Position) ANOVA was performed on selection time and direction error. The results revealed that Technique had no significant main effects on selection time ($F_{3, 45}$ = 2.096, p>.1, η^2_P = .027), but direction error ($F_{3, 45}$ = 30.477, p<.001, η^2_P = .289). Post-hoc Tukey tests showed that 3DArrow was significantly faster than 3DMinimap (p<.05) and Radar (p<.001). They also indicated that 3DWedge, 3DArrow, and 3DMinimap were more accurate than Radar (all p<.001), with no significant difference among these three.

For the "Direction" task, Target Appearance Position did not show significant main effects on selection time ($F_{2, 30}$ = 2.338, p>.05, η^2_P = .020).

5.4 User Preference

We used a questionnaire with 5-point Likert scale questions that asked participants if they enjoyed using the techniques; if they felt confident using them; and if they thought that the techniques were useful. The results show that they felt both confident (3.93) and comfortable (3.81) using 3DWedge for all four tasks, and also enjoyed (3.63) using it the most. They felt neutral about 3DArrow on average (3.19). 3DMinimap (2.73) and Radar (2.23) both received low average ratings.

Users commented that 3DWedge was particularly efficient for estimating the distance between targets and their location. For example, one participant commented that "*It was really fast and convenient to compare the distance through the square* [Wedge base]". On the other hand, participants sometimes felt it was difficult to know the direction of the 3DWedge. Without a visual element to show the middle a wedge, participants also thought it was sometimes hard to estimate the distance between two targets using the bases. 3DArrow was simple to learn, but when there was a large number of targets, it was difficult to compare the length of the sticks to get distance information. 3DMinimap helped give a fast overview of the targets' location, but participants had to move their head back-and-forth very frequently to compare target distances, which was time-consuming. They also said it was very tiring. More than half of them said that Radar was not a good technique to use.

5.5 Discussion

We found that for the "Closest to User" task, Radar was not significantly worse than 3DWedge, 3DArrow, and 3DMinimap regarding selection time and error rate–which contradicted **H.1**. This suggests that for simple tasks (for example, finding the closest target), existing 2D map-based techniques using text to show height information may still work. In addition, the results show that 3DMinimap performed worse than 3DWedge and 3DArrow. The reason, as indicated by some participants, was that there was a need to provide extra user-target indicator(s) that could help them compare distances with more ease. They found it especially difficult when the targets were far from the user's location since the targets looked like they were suspended in midair.

H.2 was not supported–the results showed that 3DArrow and 3DMinimap did not perform better than 3DWedge. However, according to the user comments, they did have difficulty estimating the distance between two targets since they could not find salient target representations that would show the exact location of the targets. It appeared that the bases of the wedges were not good enough. 3DWedge could be further improved to address the problem. The results also suggested that 3DArrow performed better than Radar. Text information was not able to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

support the "Closest Target Pair" task that well.

The results supported **H.3**: 3DWedge did perform better than all the other three techniques. 3DWedge led to significantly lower errors than 3DArrow and Radar and was significantly faster than 3DMinimap. However, we also found that in a dense environment, there were no significant differences among all four techniques. It seemed that the pyramid base helped to estimate distances between the participants' location and targets when there were not that many targets, but the base alone was not enough when there were more targets.

For the "Direction" task, 3DArrow, 3DWedge, and 3DMinimap were much more accurate than Radar (supporting **H.4**). Complementary text about height information in Radar was not that useful to support this task. Although the results suggested that participants were able to generate similar results with 3DArrow and 3DMinimap, 3DWedge led to some confusion because it is not easy to imagine the middle point of the base of the wedges. This problem may be related to the Necker cube effect [37] because there were no explicit direction indicators to aid participants in determining the center of the bases and the viewing perspective, thereby making comparisons between bases difficult.

Our results altogether indicate that Radar was not suitable for target visualization in a 3D environment. 3DWedge was the best-performing technique overall and might be sufficient for low-density scenarios and simple tasks but not for high-density and more complex tasks. 3DArrow and 3DMinimap have the potential for further development but need more radical changes.

Based on our results (users' preference and task performance), it seemed that 3DWedge had the greatest potential of all four techniques. In addition, as indicated by users, 3DWedge would need only small adjustments and modifications to enhance its usefulness when dealing with more complex tasks and also with a larger number of targets. Because of these findings, we decided to extend it by incorporating additional visual elements.

6 3DWEDGE+: ENHANCING 3DWEDGE

We wanted to make 3DWedge more applicable to higher density configurations and improve its usefulness. To achieve this, we needed to see what additional elements it was lacking and if any of the features from the other three techniques could be used.

From the first study, we know that **(1)** explicit target representation was preferred by users in the "Closest Target Pair" task and that a square base was not able to show the exact location of the targets. **(2)** An extra user-target indicator like the sticks in 3DArrow that connected the user to the targets was shown to be very useful in the "Closest to User" task. **(3)** The line was a clear and accurate direction indicator to support identifying the location of the targets from the users' viewing direction.

From the above analysis, we were able to identify the visual element that could enhance 3DWedge. First, we added a ball header to the middle of the square base of 3DWedge and linked the header and the reference point with a line (Fig. 7). Second, to help users infer distance more accurately in a dense configuration, we initially considered adding the sphere layers similar to 3DMinimap, but this feature would add unnecessary visual clutter. Instead, we added tiny evenly-distributed dash lines to the middle line connecting the ball to the base of the wedges as the distance indicator. In this way, users could simply count the number of these tiny dash lines to determine the distance. The furthest target would be on the fifth line in this experimental setting. We ran a pilot study with six participants comparing the techniques with or without the dash lines across four tasks. We found the technique with the dash lines achieved higher performance on average.

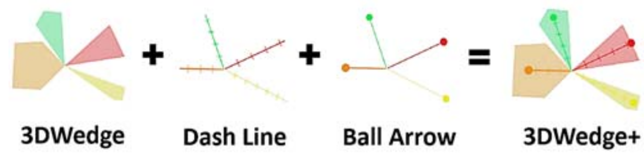


Fig. 7. The additional visual elements in the design of 3DWedge+.

We also thought about using text but went against it because, with the dash lines, text became redundant. Also, it will make the visualization too cluttered. The final version of 3DWedge+ is shown in Fig. 7.

7 STUDY 2: EVALUATING 3DWEDGE+

As 3DWedge, 3DArrow, and 3DMinimap contributed some of their features to 3DWedge+, we decided to use the three techniques as baselines and compared them with 3DWedge+. This experiment aimed to find out how well 3DWedge+ would perform in relation to the other techniques.

7.1 Hypotheses

H.5. *3DWedge+ would require more selection time than 3DWedge and 3DArrow for all tasks.* Due to the higher visual complexity of 3DWedge+, it would take users extra time to figure out the right answer than the other techniques.

H.6. *3DWedge+ would yield more accurate results than all the other techniques in the "Order" task.* Because we added the dash lines to the middle of 3DWedge+ as an additional distance indicator, this would help users distinguish even minor distances between objects more accurately.

H.7, H.8. *Users would find 3DWedge+ easier to use than 3DWedge when doing the "Closest Target Pair" (H.7) and the "Direction" (H.8) tasks*. Since we added the ball arrow as a salient target representation (which was helpful for the "Closest Target Pair" task) and the middle line as an explicit direction indicator (useful for the "Direction" task), the user would likely find these two tasks easier.

7.2 Participants, Experiment Design and Procedure

In this experiment, we recruited another 16 (13 males and 3 females) participants from the same local university. Their ages ranged from 19 to 26 (M=21). They could all recognize the colors of the targets clearly and had no vision issues. None of them had participated in experiment one.

We used the same experimental procedure and design

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

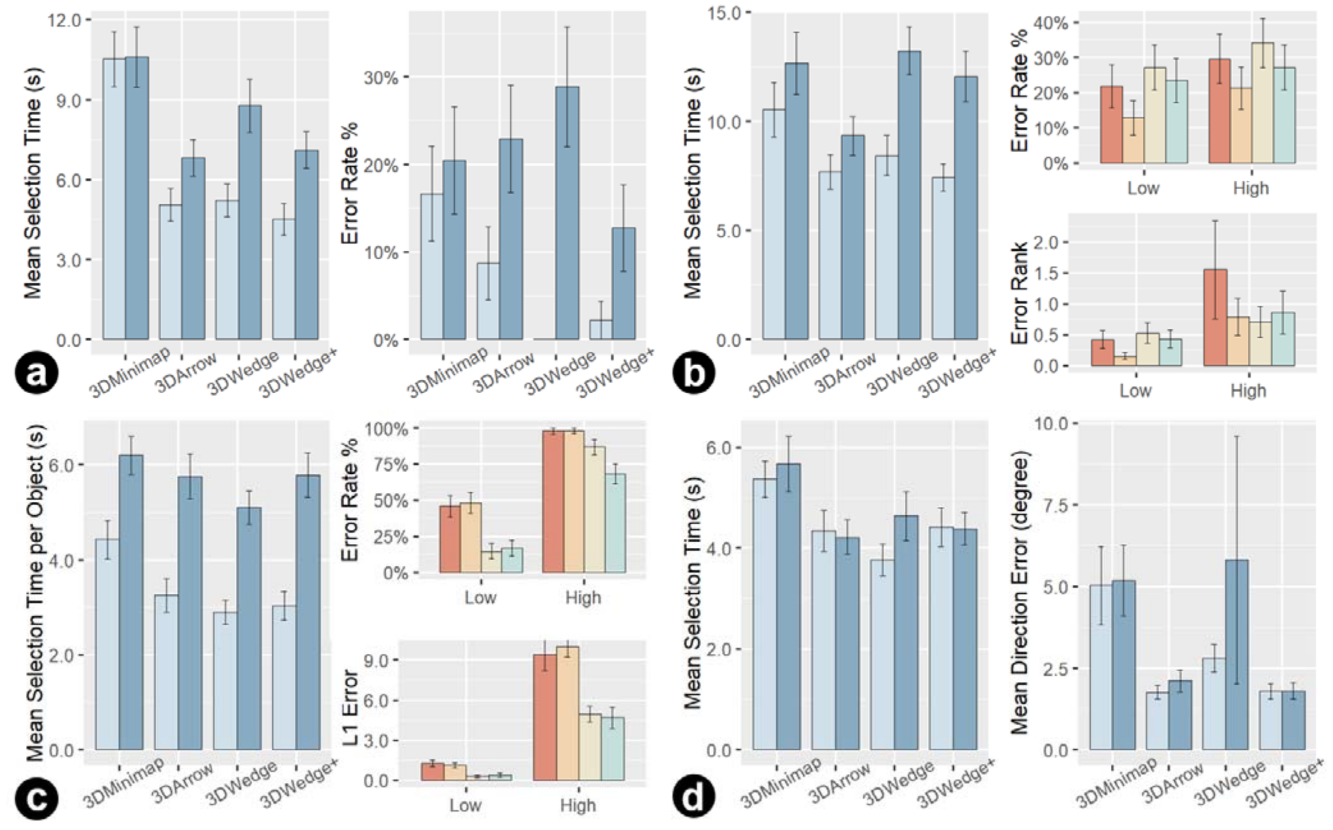


Fig. 8. (a) The “Closest to User” Task mean selection time and error rate; (b) the “Closest Target Pair” Task mean selection time, error rate and error rank; (c) the “Order” Task mean selection time per object, error rate and mean L1 Error; and (d) the “Direction” Task mean selection time and mean direction error. Error bars all indicate standard error.

as in the previous experiment, except that we replaced Radar with 3DWedge+ and repeated each trial 3 times. The study used a 4×2 within-subjects design with two factors: Technique (3DMinimap, 3DArrow, 3DWedge, or 3DWedge+) and Density (4 Targets or 9 Targets). Thus, for each task, we gathered 4 (techniques) × 2 (density levels) × 3 (blocks) × 16 (participants) = 384 timed trials.

7.3 Experimental Results

Like the previous study, the results were organized by hypotheses. Fig. 8 summarizes the performance of the four techniques based on the four tasks. Similarly, we removed outliers (mean + 3std.) for each task. We dropped 9 trials (~2.3%) for the “Closest to User” task leaving 375 trials. The “Closest Target Pair” and the “Order” tasks both dropped 10 trials (~2.6%) leaving 374 trials. The “Direction” task dropped 7 trials (~1.8%) leaving 377 trials.

For **H.5**, the ANOVA showed that Technique had a significant main effect on selection time in the “Closest to User” task ($F_{3,45} = 14.771$, $p<.001$, $\eta^2_P = .108$), the “Closest Target Pair” task ($F_{3,45} = 3.312$, $p<.05$, $\eta^2_P = .026$), and the “Order” task ($F_{3,45} = 4.001$, $p<.01$, $\eta^2_P = .032$). Post-hoc Tukey tests indicated that 3DWedge+, 3DArrow, and 3DWedge were significantly faster than 3DMinimap but had no significant difference between each other in these three tasks. Technique had no significant effect on selection time in the “Direction” task ($F_{3,45} = 1.543$, $p>.1$, $\eta^2_P = .013$).

For **H.6**, ANOVA tests showed that Technique had a significant main effect on error rate ($F_{3,45} = 15.033$, $p<.001$, $\eta^2_P = .110$) and L1 Error ($F_{3,45} = 14.383$, $p<.001$, $\eta^2_P = .105$). Post-hoc analyses showed that 3DWedge+ and 3DWedge were significantly more accurate than the other techniques (all $p<.001$). They also led to fewer large errors than 3DMinimap and 3DArrow in this task. In addition, the error rate of 3DWedge+ in the 9-target configuration was significantly lower than all the other techniques (all $p<.05$).

7.4 User Preference

In the post-experiment questionnaire, we not only asked the participants to complete a set of 5-point Likert scale questions but also asked them to select what they felt was the best technique for each task. Results indicated that users felt most confident using 3DWedge+ overall and also liked it the most. The ratings for 3DWedge+ for enjoyment (4.56), confidence (4.38), and usability (4.44) were all high.

Most users felt that 3DWedge+ was most useful for the “Closest to User” task (10/16) and “Order” task (12/16). Seven (7/16) of them felt that 3DWedge+ was the most useful for the “Closest Target Pair” task; five (5/16) felt that 3DArrow was the most useful, and only one (1/16) felt 3DWedge was the most useful. Users felt it is easiest to use 3DArrow (10/16) for the “Direction” task but thought 3DWedge+ (4/16) was more useful than 3DWedge (1/16).

7.5 Discussion and Future Work

In terms of selection time, we found that 3DWedge+, 3DWedge, and 3DArrow had similar performance. We found that even adding some extra visual features, such as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

small dash lines, would not lower the performance of the 3DWedge+, and this went against what we initially hypothesized (**H.5**).

By adding the dash lines and ball arrow to improve estimating distance information, we found that our **H.6** was supported. 3DWedge+ was significantly more accurate than the other techniques in high-density configurations in the "Order" task, while in the less dense scenarios it had similar performance to 3DWedge. Data from the post-experiment questionnaire also supported this when participants said that they felt 3DWedge+ was the most useful technique in the "Order" task. This clearly shows that the additional distance indicator represented an improvement, primarily because of the dash lines.

For the "Closest Target Pair" task, users commented that they felt more confident using 3DWedge+ than 3DWedge; they also selected it as the one they most preferred technique for the task. We also found that 3DWedge+ (25%) did improve accuracy compared to 3DWedge (31%), although the error rate was still slightly higher than 3DArrow (17%). As such, it seemed that the ball arrow did help with estimating the distance between two targets (supporting **H.7**). Furthermore, according to the user comments 3DWedge+ was more intuitive for participants to determine the direction of targets. It also improved the accuracy in the "Direction task" according to the data analysis—these two observations supported **H.8**.

When cross-checking the results of the two studies, we found some inconsistencies between them. For example, in the "Closest to User" task, the difference was found between 3DWedge and 3DMinimap in the second study, but not the first one. We think that there were three reasons for these variations. First, we collected more trials in the second study. More practice could have helped participants learn the techniques better. Second, because different users had different 3D spatial abilities, they would have worked differently with each visualization technique. And third, the randomization of the targets could have had an impact. In the future, it would be important to measure the spatial abilities of the participants and used them as a covariate in the analysis. Furthermore, a longer-term study over a wider age range of participants could be considered to evaluate the generalizability of the techniques.

Moreover, we found that 3DWedge's standard error of the mean direction error in the "Direction" task was abnormal (see Fig. 8d). A close examination of the data revealed that one participant chose the opposite direction (about 180°) when using 3DWedge. This might suggest that the Necker cube effect did indeed occur with 3DWedge which had no visual cues to help determine its orientation.

For both 3DWedge and 3DWedge+, we found that users' performance was superior on 3D visualization tasks, especially in distinguishing distances of targets, over the other techniques. 3DWedge+ was more accurate in selection across the four tasks and was significantly better in high-density configurations, while 3DWedge was more suitable in configurations with fewer targets because it was simpler and had less visual occlusion of other background elements in the 3D environment.

8 Design Recommendations

From the results of our two studies, we are able to distill the following seven recommendations for the design of visualization techniques of off-screen and occluded targets in 3D VE.

R1. *When the number of targets is relatively small, one can use simple visualizations, like 3DWedge, to help users visualize these targets*. The results of the first study show 3DWedge is competitive enough for finding the direction of targets and comparing both target-to-target and user-to-target distances. It also has less visual clutter and obstruction.

R2. *When the number of targets is large, one may need to add more visual features or elements that are supportive of the goal of the task*. For example, we added more distance indicators (the scale lines) to 3DWedge+ to help users identify the user-to-target distances more accurately. The results of the second study suggest that 3DWedge+ can better support users in distinguishing even small user-to-target distances.

R3. *Use lines to indicate or even highlight direction orientation*. Our results suggest that explicit and clear line direction, such as an arrow (3DArrow), ball arrow (3DWedge+) and center view (3DMinimap) is preferred by users to get an accurate sense of direction orientation. This is also supported by previous work [13] in 2D.

R4. *Use a clear visual element to represent the targets.* In both studies, the users have preferred a precise target representation (such as the arrow tip and the small sphere) rather than an approximation (square base). The target-to-target distance estimation accuracy of 3DWedge is also shown to be improved with a clear visual indicator in the second study.

R5. *Use a non-linearly scaled visual feature to compare minor user-to-target distances*. Our results indicate that the non-linearly scaled base of the 3DWedges significantly increases its accuracy in comparing the minor distances between targets and the user. Designers can take these non-linear equations into account when there is a need for distinguishing minor distances.

R6. *Use evenly-distributed dash lines to make the distance comparison even more precise*. The second study reveals that the dash lines allow user-to-target distance comparisons more accurate even in the high-density configurations.

R7. *Caution is needed when reducing visual occlusion since it could lead to ambiguous representations in the 3D environment*. Deleting all lines in 3DWedge has helped reduce the visual clutter. However, as suggested by the results, it might also induce the Necker cube effect which causes ambiguity in people's visual perception of 3D objects.

9 Conclusion

In this paper, we have presented five target visualization techniques (3DWedge, 3DArrow, 3DMinimap, Radar, and 3DWedge+) that can support users with location awareness of off-screen and occluded targets in 3D virtual reality environments. These techniques help users determine the direction of the targets and compare target-to-target and user-to-target distances. We also have developed a framework that can be used for assessing the performance of these techniques. This framework includes four main tasks (three distance tasks and one direction task) and their assessment criteria. Our two user studies have evaluated each technique's performance through these four tasks with two levels of density configurations. Overall, our results indicate that 3DWedge is a simple and accurate technique for visualizing off-screen and occluded targets in 3D virtual reality environments while 3DWedge+ is suitable for complex tasks that deal with a larger number of targets and need higher level accuracy and precision.

In all, we believe the development of 3DWedge and 3DWedge+ is an important step toward improving the visualization of off-screen and occluded targets in 3D head-mounted display based virtual environments. In addition, from the two experiments, we have extracted recommendations which can help the design of other visualization techniques for these 3D virtual environments.

Acknowledgment

We would like to thank all the participants for their time and the reviewers for their valuable comments and suggestions that have helped us improve our paper. This research has been partially funded by Xi'an Jiaotong-Liverpool University (XJTLU) Key Special Fund (#KSF-A-03) and XJTLU Research Development Fund.

Hai-Ning Liang is the corresponding author and can be reached via email at HaiNing.Liang@xjtlu.edu.cn.

References

[1] P. Baudisch and R. Rosenholtz, "Halo: A technique for visualizing off-screen locations," *Proc. CHI*, pp.481–488, 2003.
[2] S. Burigat, L. Chittaro and S. Gabrielli, "Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches," *Proc. MobileHCI*, pp.239–246, 2006.
[3] L. Chittaro and S. Burigat, "3D location-pointing as a navigation aid in Virtual Environments," *Proc. AVI*, pp.130-137, 2004.
[4] H. Jo, S. Hwang, H. Park and J. Ryu, "Aroundplot: focus+context interface for off-screen objects in 3D environments," *J. Computers and Graphics*, vol.35, no.4, pp.841-853, May 2011.
[5] R.A. Ruddle, S.J. Payne and D.M. Jones, "Map usage in virtual environments: orientation issues," *Proc. IEEE*, pp.133-140, 1999.
[6] A.J. Aretz, "The design of electronic map displays," *J. Human Factors*, vol.33, no.1, pp.85-101, Feb. 1991.
[7] R. Stoakley, M.J. Conway and R. Pausch, "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. CHI*, pp.265-272, 1995.
[8] J. Mackinlay, L. Good, P. Zellweger, M. Stefik, and P. Baudisch, "City Lights: Contextual views in minimal space," *Proc. CHI*, pp.838–839, 2003.
[9] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani, "Wedge: clutter-free visualization of off-screen locations," *Proc. CHI*, pp.787–796, 2008.
[10] S. Gustafson and P.P. Irani, "Comparing visualizations for tracking off-screen moving targets," *Proc. CHI Extended Abstracts*, pp.2399–2404, 2007.
[11] Z. Hossain, K. Hasan, H.N. Liang, and P. Irani, "EdgeSplit: Facilitating the selection of off-screen objects," *Proc. MobileHCI*, pp.79-82, 2012.
[12] A. Ion, B.Y.-L. Chang, M. Haller, M. Hancock, and S.D. Scott, "Canyon: providing location awareness of multiple moving objects in a detail view on large displays," *Proc. CHI*, pp.3149–3158, 2013.
[13] D. Miau and S. Feiner, "Personalized Compass: A compact visualization for direction and location," *Proc. CHI*, pp.5114–5125, 2016.
[14] S. Burigat and L. Chittaro, "Visualizing references to off-screen content on mobile devices: a comparison of Arrows, Wedge, and Overview+Detail," *J. Interacting with Computers*, vol.23, no.2, pp.156-166, Mar. 2011.
[15] T. Schinke, N. Henze, and S. Boll, "Visualization of off-screen objects in mobile augmented reality," *Proc. MobileHCI*, pp.313–316, 2010.
[16] T. Siu and V. Herskovic. "SidebARs: Improving awareness of off-screen elements in mobile augmented reality," *Proc. ChileCHI*, pp.36–41, 2013.
[17] P. Perea, D. Morand, and L. Nigay "Halo3D: a technique for visualizing off-screen points of interest in mobile augmented reality," *Proc. ISMAR-Adjunct*, 2017.
[18] M. Trapp, L. Schneider, N. Holz, and J. Döllner, "Strategies for visualizing points-of-interest of 3D virtual environments on mobile devices," *Proc. The sixth international symposium on LBS & TeleCartography*, 2009.
[19] L. Chittaro and S. Venkataraman, "Navigation aids for multi-floor virtual buildings: a comparative evaluation of two approaches," *Proc. VRST*, pp.227–235, 2006.
[20] S. Burigat, L. Chittaro, and A. Vianello, "Dynamic visualization of large numbers of off-screen objects on mobile devices: an experimental comparison of Wedge and overview+detail," *Proc. MobileHCI*, pp.93–102, 2012.
[21] P.M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *J. Journal of Experimental Psychology*, vol.47, no.6, pp.381–391, Jun. 1954.
[22] R.A. Ruddle, S.J. Payne, and D.M. Jones, "The Effects of Maps on Navigation and Search Strategies in Very-Large-Scale Virtual Environments," *J. Journal of Experimental Psychology: Applied*, vol.5, no.1, pp.54-75, Mar. 1999.
[23] M. Sarkar, S.S. Snibbe, O. J. Tversky, and S.P. Reiss, "Stretching the rubber sheet: a metaphor for viewing large layouts on small screens," *Proc. UIST*, pp.81–91, 1993.
[24] R. Brath, "3D InfoVis is here to stay: deal with It," *Proc. IEEE VIS*, pp.25–31, 2014.
[25] M. Saenz, A. Baigelenov, Y.H. Hung, and P. Parsons "Reexamining the cognitive utility of 3D visualizations using augmented reality holograms," *Proc. IEEE VIS*, 2017.
[26] T. Gonçalves, A.P. Afonso, M.B. Carmo, and P. Pombinho, "Comparison of off-screen visualization techniques with representation of relevance on Mobile Devices," *Proc. BCS-HCI*, 2014.
[27] L. Chittaro and I. Scagnetto, "Is Semitransparency Useful for Navigating Virtual Environments?" *Proc. VRST*, pp.159–166, 2001.
[28] A. Fedosov and S. Misslinger. "Location based experience design for mobile augmented reality," *Proc. EICS*, pp. 185-188. 2014.
[29] T. Hakala, J. Lehikoinen, and A. Aaltonen. "Spatial interactive visualization on small screen," *Proc. MobileHCI*, pp. 137-144, 2005.
[30] K. Hartmann, T. Götzelmann, K. Ali, and T. Strothotte. "Metrics for functional and aesthetic label layouts," *Proc. International Symposium on Smart Graphics*, pp. 115-126, 2005.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2019.2905580, IEEE Transactions on Visualization and Computer Graphics

[31] D. Bowman, E. Kruijff, J.J. LaViola Jr, and I.P. Poupyrev, *3D User interfaces: theory and practice*, Addison-Wesley, 2004.

[32] A. Cockburn, A. Karlson, and B.B. Bederson. "A review of overview+detail, zooming, and focus+context interfaces." *J. ACM Computing Surveys*, vol. 41, no. 1, 2009.

[33] U. Gruenefeld, A.E. Ali, W. Heuten, and S. Boll. "Visualizing out-of-view objects in head-mounted augmented reality." *Proc. Mobile HCI*, p. 81, 2017.

[34] M. Sukan, C. Elvezio, O. Oda, S. Feiner, and B. Tversky. "Parafrustum: Visualization techniques for guiding a user to a constrained set of viewing positions and orientations." *Proc. CHI*, pp. 331-340, 2014.

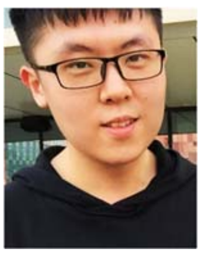
[35] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. "Feature congestion: a measure of display clutter." *Proc. CHI*, pp. 761-770, 2005.

[36] F. Biocca, A. Tang, C. Owen, and F. Xiao. "Attention funnel: omnidirectional 3D cursor for mobile augmented reality platforms." *Proc. CHI*, pp. 1115-1122, 2006.

[37] L.A. Necker. "Observations on some remarkable optical phenomena seen in Switzerland." *J. London and Edinburgh Philosophical Magazine and Journal of Science*, pp. 329-37, 1832.

[38] S. Matsuzoe, S. Jiang, M. Ueki, and K. Okabayashi. "Intuitive visualization method for locating off-screen objects inspired by motion perception in peripheral vision." *Proc. AH*, pp. 29, 2017.

Difeng Yu received his BSc degree in Computer Science from Xi'an Jiaotong-Liverpool University (XJTLU) in 2018. He is currently a research assistant at VR Lab at XJTLU. His research interests center around the field of human-computer interaction.



Kaixuan Fan is an undergraduate student at Xi'an Jiaotong-Liverpool University, majoring in Computer Science. He has interests in virtual reality and eye-tracking technologies.

Charles Fleming is an Associate Professor in the Department of Computer Science and Software Engineering at Xi'an Jiaotong-Liverpool University. He has an undergraduate degree in mathematics and a PhD in Computer Science from the University of California Los Angeles.

Hai-Ning Liang is a Senior Associate Professor in the Department of Computer Science and Software Engineering at Xi'an Jiaotong-Liverpool University. He obtained his PhD in Computer Science from Western University and was previously a Postdoctoral Fellow at University of Manitoba and a Researcher at National ICT Australia. His research interests are in the areas of human-computer interaction, information visualization, and virtual/augmented reality technologies.

Heng Zhang received his BEng degree in Telecommunications Engineering from Xi'an Jiaotong-Liverpool University (XJTLU) in 2018. His research interests include wireless systems and human-computer interaction.

Konstantinos Papangelis is a Lecturer (Assistant Professor) in the Department of Computer Science and Software Engineering at Xi'an Jiaotong-Liverpool University. His research focuses on location-based social networks, the physical web, location-based/in-situ crowdsourcing, pervasive games, and novel mobile technologies.