Towards Efficient Visual Guidance in Limited Field-of-View Head-Mounted Displays

Felix Bork, Christian Schnelzer, Ulrich Eck, and Nassir Navab

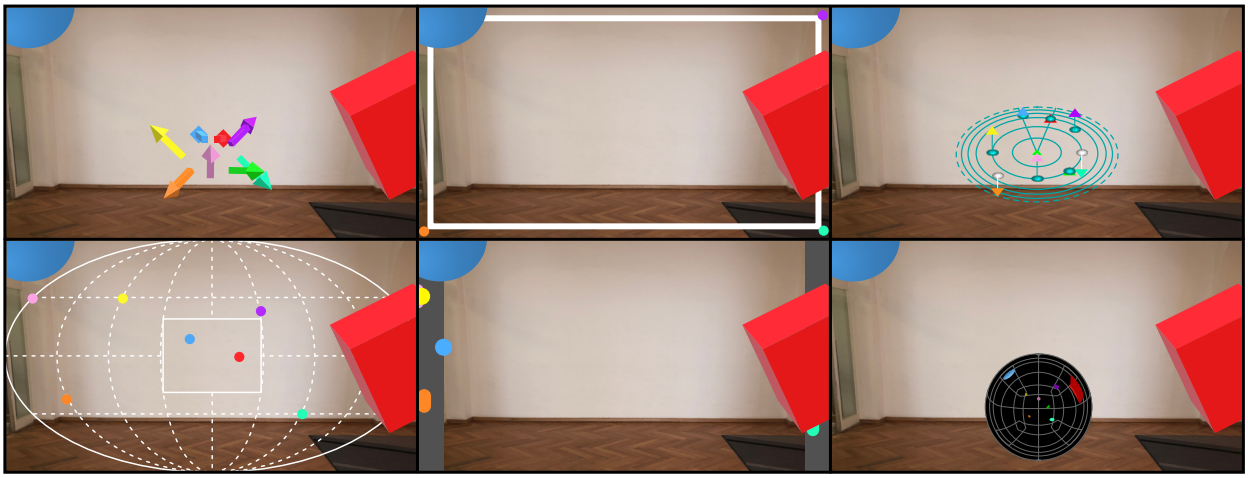

Fig. 1. Screenshots of the six different visual guidance techniques which were compared with respect to their potential for displaying out-of-view objects in Mixed Reality environments: 3D Arrows (top-left), AroundPlot (top-center), 3D Radar (top-right), EyeSee360 (bottom-left), sidebARs (bottom-center), and Mirror Ball (bottom-right).

Abstract—Understanding, navigating, and performing goal-oriented actions in Mixed Reality (MR) environments is a challenging task and requires adequate information conveyance about the location of all virtual objects in a scene. Current Head-Mounted Displays (HMDs) have a limited field-of-view where augmented objects may be displayed. Furthermore, complex MR environments may be comprised of a large number of objects which can be distributed in the extended surrounding space of the user. This paper presents two novel techniques for visually guiding the attention of users towards out-of-view objects in HMD-based MR: the 3D Radar and the Mirror Ball. We evaluate our approaches against existing techniques during three different object collection scenarios, which simulate real-world exploratory and goal-oriented visual search tasks. To better understand how the different visualizations guide the attention of users, we analyzed the head rotation data for all techniques and introduce a novel method to evaluate and classify head rotation trajectories. Our findings provide supporting evidence that the type of visual guidance technique impacts the way users search for virtual objects in MR.

Index Terms—Mixed / Augmented reality, Visualization design and evaluation methods

1 Introduction

Mixed Reality (MR) environments present both real and virtual world objects simultaneously on a single display [30]. One major advantage of such augmented environments is that the 3D location of virtual-world objects is commonly known. Therefore, efficient techniques for guiding the attention of users towards such objects can be developed, which requires both tracking and detection methods for real-world objects. Providing the user with positional information about all virtual objects enables quick exploration and navigation of MR environments as well as precise and goal-oriented actions.

One challenging task that remains an open research problem is how to convey information about surrounding virtual objects to the user. MR environments can potentially comprise a large number of virtual objects at various locations, making it difficult to understand and navigate an augmented scene. This problem is further enhanced by the fact that current Head-Mounted Displays (HMDs) still have a limited field-of-view which does not resemble the one of the human visual system. Therefore, only a small portion of the virtual environment is visible and many virtual objects are likely to be out-of-view. Several visualization techniques have been proposed in the past to locate and guide attention towards such out-of-view objects in both mobile and HMD-based MR environments. However, none of the proposed methods universally solves the problem. Drawbacks of existing methods include taking up large amounts of display real-estate, visual clutter, and occlusion issues for a large number of objects. Additionally, existing studies often lack a fundamental analysis on *how* participants utilize a certain visual guidance technique for the task of localizing virtual objects. In our opinion, a crucial building block in developing such an understanding is the analysis of head rotation trajectory data and the influence of different visualization techniques on this data.

Our contributions in this paper are two-fold: first, we introduce two novel methods for visualizing out-of-view objects in MR environments

- *Felix Bork is with Technical University of Munich, Germany. E-mail: felix.bork@tum.de.*
- *Christian Schnelzer is with Technical University of Munich, Germany. E-mail: christian.schnelzer@tum.de.*
- *Ulrich Eck is with Technical University of Munich, Germany. E-mail: ulrich.eck@tum.de.*
- *Nassir Navab is with Technical University of Munich, Germany. E-mail: nassir.navab@tum.de.*

Manuscript received 19 Mar. 2018; accepted 10 July. 2018.
Date of publication 14 Sept. 2018; date of current version 28 Sept. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2868584

and evaluate them in a user study. The first one is the Mirror Ball, which resembles a reflective sphere and displays distorted reflections of virtual objects according to their location in the environment. The second method we introduce is the 3D Radar, a technique that is often used in commercial computer games, especially in space simulators. Similar to traditional top-down mini-maps and radars, virtual objects are substituted by a small proxy icon on the radar. The position on the 2D radar plane reflects the horizontal angle from the user, while the vertical angle is encoded by orthogonal lines from the 2D radar plane towards the object icon. We compared our proposed methods with current state-of-the-art techniques for visualizing out-of-view objects during a user study with twenty-four participants. We measured task completion times in three different object collection scenarios that resemble real-world exploratory and goal-oriented visual search tasks. The 3D Radar was found to yield comparable results to state-of-the-art visual guidance techniques, while simultaneously covering a much smaller portion of the field-of-view than comparable methods. The Mirror Ball faced several perceptual challenges which resulted in higher mental effort scores and slower task completion times.

Our second contribution is a new area-based method for evaluating and classifying head rotation trajectory data. Our algorithm can be employed to discriminate between two distinct object targeting approaches in HMD-based MR environments: a direct, one-way approach following the shortest path between a start and target orientation vs. an indirect, two-way approach which aligns the horizontal and vertical angle sequentially in an L-shaped fashion. Our results indicate, that the 3D Radar encourages the latter class of trajectories, while users predominantly chose the direct, one-way approach for all other visualization methods.

2 Related Work

2.1 Visualization Techniques for Out-of-view Objects

Existing techniques for visualizing out-of-view objects in 3D Mixed Reality environments can be grouped into the following three categories: *Overview & detail* describes a category of approaches where two separate windows are used, which are usually displayed on top of each other [33]. The overview window typically conveys information about the users' surroundings, e.g. in form of a map. The detail window provides specific information about a local sub-space, typically what the user is currently looking at. In *Focus & Context* approaches there is only one window which provides a distorted view of the user's surrounding, e.g. using a fisheye projection [13, 19]. The transition between the non-distorted focus area, typically in the middle of the window, and the distorted context area is smooth. *Contextual Views* [19, 22] as the third category of visualization approaches overlay the detail window with abstract indicators, such as arrows, that provide information about the location of out-of-view objects.

Out-of-view object visualization techniques can be further classified based on the dimensionality of their information encoding. While 3D techniques provide information that enables the user to infer the location of virtual objects in 3D space, information conveyance for 2D techniques is limited to a left-or-right and above-or-below discrimination and lacks information about the depth of objects relative to the user. Numerous 2D techniques have been proposed in the past, most of which were designed specifically for desktop or mobile applications. City Lights was developed by Zellweger et al. and uses highlights at the edges of the screen to indicate the position of out-of-view windows [45]. Baudisch et al. introduced the Halo technique, which is capable of encoding the position of arbitrary out-of-view objects by drawing circles with distance-encoding radii around the object locations [8]. The circles intersect with the screen, providing the user with information about the direction and distance of objects. However, both techniques are difficult to interpret in the presence of multiple objects. Gustafson et al. proposed Wedges, where triangle shapes span from the objects' location to the edges of the screen [19]. EdgeRadar [20] and 2D Arrows [12] also take advantage of screen edges by displaying either dots or arrows respectively, that indicate the location of out-of-view objects. Siu and Herskovic picked up on these ideas and introduced the sidebARs technique, which displays vertical bars at the left and right edges of the screen [36]. Small proxy icons representing the out-of-view objects are placed inside those bars and their position within the bars indicates the position of the objects. In case multiple objects are positioned in close proximity to each other, occlusions and edge clutter are two common disadvantages of the previous techniques.

In HMD-based Mixed Reality, especially for tasks such as exploring, navigating, and achieving certain goals inside an augmented environment, 2D techniques for visualizing out-of-view objects are insufficient. Therefore, a number of techniques have been introduced that extend the information conveyance about the location of such objects to 3D space. 3D Halo and 3D Halo Projection are natural extensions of the previously mentioned Halo technique [39]. Rather than using 2D circles, the technique works by placing 3D rings or spheres around the objects. Another extension from 2D to 3D is employed by the 3D Arrows technique, which displays arrows pointing towards the 3D position of out-of-view objects [35]. For these techniques, the same drawbacks which are present in the 2D case (clutter and occlusions) also apply in 3D. Parafrustum [37] and Attention Funnel [10] are two techniques which both use a tunnel-like visualization to guide the user towards a single object. A different approach is AroundPlot, which uses a rectangle to indicate the focus area of the view [22]. The 3D space around the user is mapped to the context area outside of the rectangle using 2D orthogonal fisheye coordinates. This techniques suffers from the corner-density problem, as a large 3D space is encoded in the areas close to the four corners of the rectangle. Gruenefeld et al. introduced a similar approach with EyeSee360 [17, 18]. They use an ellipse that spans across the entire screen in combination with a smaller rectangle centered in the middle of the screen. The latter marks the focus area, while the area between the rectangle and the ellipse encodes the 3D space around the user. Multiple smaller ellipses and horizontal lines are used as helplines. EyeSee360 takes up a large portion of the field-of-view, which potentially degrades the perception of real world objects. Recently, Renner and Pfeiffer introduced the SWAVE technique, which employs wave-like motions to guide the user's attention towards a target object [34]. In the same paper, the authors mention another approach which uses screen flickering at different frequencies to indicate how close the user is to the target. A similar techniques was proposed by Matsuzoe et al., who used circular, vibrating icons at the edges of the screen [28]. However, these techniques perform best for single objects and would suffer from severe clutter in case of more complex MR environments.

Using a Mirror Ball to visualize objects that are not covered by the field-of-view has been used previously in traditional 3D modeling applications by McCrae et al. [29]. In their approach, objects are clustered based on their distance to the ball, which is divided into separate areas, similar to a Voronoi diagram. Each partition area of the ball reflects a group of objects that have a similar distance to the ball. This way, all objects appear at about the same size on the ball, independent of their distance. A drawback of this method is that the distance information between the objects and the mirror ball is lost completely. The 3D Radar is a minimap-like visualization technique that is heavily inspired by commercial computer games. Especially space simulator games such as Elite:Dangerous [1], Eve:Valkyrie [2] and Star Citizen [3] employ this technique for providing the player with an overview of the 3D space around him. It extends the traditional 2D or top-down radar with the ability to visualize objects anywhere in 3D space. While radar visualizations have been proven to be useful in 2D applications [11], to the best of our knowledge the extension to 3D space has not been evaluated in the literature yet.

2.2 Head Rotation Trajectory Analysis

In order to get a better understanding of how the different visualizations are utilized for the task of targeting virtual objects, we were interested in analyzing the head rotation trajectories. The primary goal of this analysis was to find out whether the head trajectory data reveals insights as to whether there exists an *optimal* class of trajectories, which is predominantly used to target virtual objects in MR, and whether visual guidance techniques that favor such potentially optimal trajectories generally allow for faster object localization. In a traditional sense,

a trajectory is defined as the path of a moving object, consisting of a set of consecutive positions in space as a function of time. In our scenario, we also use the notion that a given head rotation path defines a trajectory by transforming the data into spherical coordinates and referring to the individual positions of the trajectory as timestamped tuples of both the horizontal and vertical angle.

The analysis of trajectory data is a broad research topic with applications in many different domains, such as activity recognition, surveillance security, anomaly detection or traffic monitoring. A recent survey of trajectory analysis techniques, covering both supervised and unsupervised techniques for trajectory clustering and classification, is provided by Bian et al. [9] To compare two trajectories, multiple similarity measures with varying complexity have been proposed in the past, most of which are distance-based measures. The most common metrics include the Euclidean distance [23], Dynamic Time Warping [31], Least Common Sub-Sequences [40], the Fréchet distance [4], and Edit distances [14]. Anagnostopoulos et al. introduced the concept of Minimum Bounding Rectangles (MBRs) to approximate the distance calculations for sub-trajectories [5]. A comparative study on the effectiveness of some of these measures is provided by Wang et al. [42]. Distance-based measures are a powerful tool for comparing similar trajectories over time, but they define only a global measure and don't take into account, that trajectories which have the same distance measure can describe very different targeting approaches towards a virtual object. Region-based measures split the domain into a grid structure and try to classify trajectories based on the identification of homogeneous grid cells, which contain only a certain class of trajectories [26]. Such region-based approaches can be very robust for classes of trajectories that are very similar, but their performance decreases if there is a large variability in the trajectory data, as expected for our head rotations. Other comparison metrics include curvature [16], significant changing points [6, 7], and discontinuities [15].

For our head rotation analysis, we were especially interested in discriminating between two distinct types of trajectories: *i)* those that correspond to a direct, one-way targeting approach, which is characterized by a straight line between a starting head rotation and the orientation of the target point in spherical coordinates; and *ii)* trajectories that describe an indirect, two-way targeting approach, which is defined by a (potentially rotated) L-shaped path in spherical coordinates. We were therefore able to define the *optimal* trajectories in both cases and applied an area-based trajectory classification approach, which is explained in more detail in section 3.3.

3 Methods

We want to compare the performance of our two proposed visualization techniques to the most promising approaches from the literature in the context of HMD-based MR environments. For this purpose, we chose three different object collection tasks that reflect real-world exploratory and goal-oriented search scenarios. We implemented an application for the Microsoft HoloLens, where participants were asked to collect a number of virtual objects, positioned around them in 3D space and outside of their field-of-view, as fast as possible for all three scenarios. The different visual guidance techniques were used to help participants find the virtual objects. In this section, we present a description about the process of selecting the set of visualization techniques which was included in our comparison. Furthermore, we provide implementation details for all techniques as well as a detailed description about our head rotation trajectory classification algorithm.

3.1 Selecting Techniques for Comparison

Since we want our subjects to collect multiple out-of-view objects during the experimental user study, Parafrustum and Attention Funnel are not applicable due their limitation of only supporting the attention guidance for a single object. The same applies for SWAVE and the two screen flickering approaches [28, 34]. We also discard 3D Halo and 3D Halo Projection since they suffer from severe visual clutter in the presence of multiple objects [17, 39]. EyeSee360 was developed with EdgeRadar as an inspiration and has been shown to outperform Halo, 2D Arrows and Wedge. Thus, we only include the superior EyeSee360 technique in our comparison and discard the other 2D methods. From the remaining 3D visualizations, we include both AroundPlot and 3D Arrows. Additionally, we include a slightly modified version of the sidebARs technique, which allows the encoding of locations in 3D space by horizontally moving the proxy icons according to their location. To the best of our knowledge, there exist no comparison between these techniques, except for AroundPlot and 3D Arrows, which have been shown to perform comparably for a small number of objects [22]. Figure 1 shows a side-by-side view of the six different visualization techniques which were compared during our user study.

3.2 Method Implementation Details

All six visualization techniques were implemented as part of a Unity 3D application to run on the HoloLens. In case modifications were made to the original implementations, they are highlighted and motivated in the following paragraphs.

3.2.1 3D Arrows

We implemented the 3D Arrows visualization similar to previous works [35]. All arrows are placed around a sphere that is positioned in front of the user. Each arrow points in the direction of a virtual object, with the length of the arrow scaled according to the distance of the object. The only difference to the original implementation is that we color the arrows based on the color of the target objects.

3.2.2 sidebARs

In the original 2D sidebARs method, the position of small proxy icons inside the two vertical bars at the edges of the screen indicates where an object is placed relative to the current position of the user [36]. While the vertical position inside the bars indicates whether an object is above or below the user, an object can only be placed either inside the left or right bar. In our implementation, we also vary the horizontal position of proxy icons inside the left and right sidebar to indicate the horizontal angle of a target object. Additionally, we removed the numeric distance indicator which is overlaid on top of the proxy icons in the original method.

3.2.3 AroundPlot

Our implementation of AroundPlot is very similar to the original method [22]. Spherical coordinates are mapped to orthogonal fish-eye coordinates, such that the visible field-of-view corresponds to a rectangular area on the screen. Virtual objects inside the MR scene are indicated as colored dots positioned around the sides of the rectangle according to their horizontal and vertical angle. Compared to the original method, we exclude the dynamic magnification feature, which magnifies the rectangle in the direction the user is moving. Instead, the size of the rectangle is fixed and covers almost the same area as the field-of-view of the HoloLens.

3.2.4 EyeSee360

For our EyeSee360 implementation, we used a version with helplines and a rectangular inner window that the authors also proposed for their implementation on Google Cardboard [17]. However, in our implementation the dots representing out-of-view objects have the same color as the objects that they correspond to as opposed to distance-encoding colors in one variant of the original method. Dots positioned inside the inner rectangle indicate objects which are covered by the field-of-view. All other objects are positioned according to their horizontal and vertical angle inside the ellipse. This way, dots on the left and right edges of the ellipse correspond to objects directly behind the user, while dots on the upper and lower edges refer to objects directly above or below the user.

3.2.5 Mirror Ball

The implementation for the Mirror Ball technique is aimed at mimicing the nature of a real-world spherical mirror. Unlike McCrea et al., [29] who use a mirror ball for navigation purposes in a traditional virtual 3D environment, we do not divide the sphere in different regions that correspond to clusters of object at different distances. We position a

virtual sphere in front of the user that has a reflective shader attached to it. In order to provide the user with a better sense of depth, we employ a cube map that has a grid texture on each side as a skybox and reflect it on the sphere. Similar to a real-world spherical mirror, the reflections visible on our Mirror Ball of all virtual objects as well as the skybox are spherically distorted. These distortions provide positional hints to the user.

3.2.6 3D Radar

The design and implementation of our 3D Radar is heavily inspired by commercial video games. The 3D Radar consists of multiple concentric circles that are placed in front of the user and tilted slightly in order to prevent perceptual occlusion issues. A small triangle is placed in the center of the radar to indicate the users' position, while a forward-facing cone represents the field-of-view. Every virtual object inside the MR scene is depicted on the radar as a combination of a proxy icon and a circle corresponding to the objects' projection in the 2D radar plane. Similar to traditional 2D (top-down) radars, the projection indicates the horizontal angle between the target object and the user. To provide the user with additional information about the vertical angle, proxy icons in the form of colored triangles located directly above or below the projection circles are used. To quickly convey information about correspondences between proxy icons and their projection circle, a line connecting the two is employed. The color of both the line and the projection circle depends on whether the virtual object is located above or below the user, cf. Figure 1.

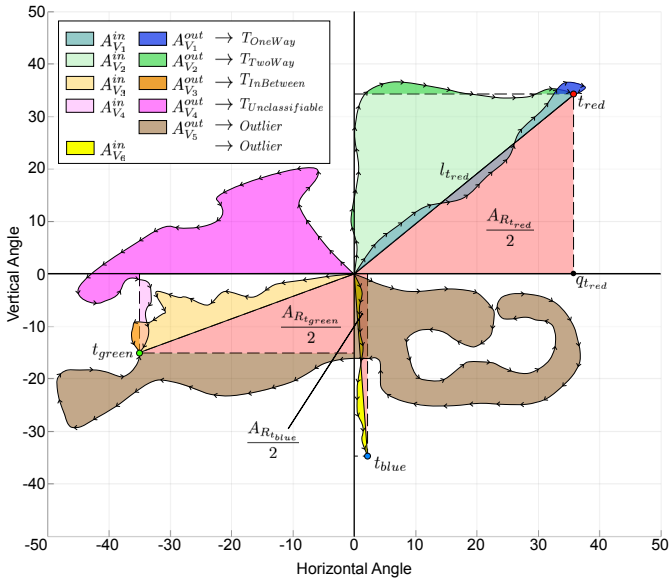

Fig. 2. Visual explanation of the areas A_V^{in} and A_V^{out} for six exemplary targeting trajectories V_1 (blue, classified as T_{OneWay}), V_2 (green, classified as T_{TwoWay}), V_3 (orange, classified as $T_{InBetween}$), V_4 (pink, classified as $T_{Unclassifiable}$), V_5 (brown, classified as $Outlier$ due to long path), and V_6 (yellow, classified as $Outlier$ due to horizontal angle of $t_{blue} \approx 0$).

3.3 Head Rotation Trajectory Classification

Our analysis for comparing the head rotation trajectories of the different visualizations is formulated as a classification problem. Our approach is driven by the hypothesis, that there are *two* very distinct ways of targeting a virtual object: the first one can be described as a direct, one-way targeting approach (T_{OneWay}), where the trajectory matches the horizontal and vertical angle of the target object simultaneously. The second way is an indirect, two-way targeting approach (T_{TwoWay}), such that the trajectory matches the two angles sequentially in an L-shaped fashion, i.e. the horizontal angle of the target object is matched first, then the vertical one (or vice versa). In spherical coordinates, a perfect one-way targeting approach towards a target point t is given by the straight line l_t connecting both the target point and the starting point s (i.e. initial head rotation). On the other hand, a perfect two-way approach is given by two connected straight lines forming the two legs of a right triangle defined by s, t, and the point on the x-axis $q = (t_x, 0)$ (or alternatively the point on the y-axis $q' = (0, t_y)$). In this way, the first line matches the horizontal angle of t, while the second line subsequently matches the vertical angle (or vice versa). Figure 2 illustrates these two targeting approaches in the first quadrant. The blue trajectory resembles a one-way targeting approach closely following the line between the origin and t_{red}, while the green trajectory resembles a two-way targeting approach.

A given head rotation trajectory can be described as a polygonal path V, consisting of a set of vertices v_i which correspond to discrete angular measurements, such that $V = \{s, v_1, v_2, ..., t\}$. Our classification approach for such polygonal paths V is based on the hypothesis, that the previous two object targeting approaches T_{OneWay} and T_{TwoWay} can be discriminated based on the area of the region enclosed by l_t and V. Assuming that the starting point s is translated to the origin, a one-way approach is characterized by an area $A_V \approx 0$, while for a two-way approach $A_V \approx \frac{A_{R_t}}{2}$, with A_{R_t} as the area of the rectangle R_t defined by the origin and t. To avoid misclassifications of polygonal paths with large portions of their enclosing area A_V lying outside of R_t, we further split A_V into the two parts A_V^{in} and A_V^{out}, such that $A_V = A_V^{in} + A_V^{out}$ and $A_V^{in} = V \cap R_t$. Two percentage thresholds α and β for A_V^{in} and A_V^{out} respectively can be defined to allow for some marginals. The final trajectory classification with two more classes for unclassifiable trajectories can now be defined as follows:

$$f(V,t) = \begin{cases} T_{OneWay}, & \text{for } A_V^{in} < \alpha, A_V^{out} < \beta \\ T_{TwoWay}, & \text{for } A_V^{in} > 100 - \alpha, A_V^{out} < \beta \\ T_{InBetween}, & \text{for } 100 - \alpha \geq A_V^{in} \geq \alpha, A_V^{out} < \beta \\ T_{Unclassifiable}, & \text{for } A_V^{out} \geq \beta \end{cases} \quad (1)$$

Figure 2 illustrates the different area definitions by depicting six polygonal paths that define different trajectories towards three target points, as well as their classification results.

The computation of the individual areas is based on splitting a polygonal path into several sub-polygons V_i, each limited by two consecutive intersections with the line l_t. The different line segments of the path are traversed and tested for intersection with l_t. In case of an intersection, all vertices starting from the previous intersection point to the current one define a sub-polygon V_i (for the first sub-polygon V_1, the origin is set to be the initial intersection point). Each time a sub-polygon is closed this way, the total area A_{V_i} of the sub-polygon is computed. Subsequently, the intersection area A_V^{in} between the sub-polygon and the rectangle R_t is computed via polygon clipping [38, 43]. Algorithm 1 provides more details on the implementation.

One minor limitation of our classification algorithm is that for the two-way targeting approach, no distinction is currently made whether the vertical or the horizontal angle is matched first. Furthermore, the classification might be imprecise for target points located very close to the coordinate axes $x = 0$ or $y = 0$, where the area of R_t is very small and can even be 0 for target points lying exactly on one of the axes. In these cases, a classification is not meaningful since one-way and two-way targeting approaches are almost the same and cannot be distinguished. In addition to that, large values for A_V^{out} and therefore misclassification are very likely to occur. To account for these cases, it is possible to simply discard these trajectories with target points close to one of the axes and classify them as outliers. The drawback of this approach is that for these cases, target-oriented trajectories are treated the same as random trajectories. However, we argue that for our research question, which is to identify patterns in the head rotation data, this can be tolerated. Another set of outliers are those trajectories, whose length exceeds a certain percentage of l_t, i.e. a trajectory that is three times longer than the optimal path is discarded and classified as

Algorithm 1 Computation of the areas A_V^{in} and A_V^{out} enclosed between a given polygonal path and the line towards the target point.

procedure CALCULATEENCLOSINGAREAS
  Input:
  $t \leftarrow$ target point
  $V \leftarrow$ list of vertices of polygonal path towards target point
  Output:
  $A_V^{in} \leftarrow$ total area inside R_t
  $A_V^{out} \leftarrow$ total area outside of R_t
  Algorithm:
  $R_t \leftarrow$ rectangular polygon defined by origin and t
  $l_t \leftarrow$ line segment between origin and t
  $V_c \leftarrow$ empty list of vertices for current sub-polygon
  for all vertices v in V **do**
    $l \leftarrow$ current line segment defined by v and $v+1$
    intersects, p $\leftarrow$ testIntersection(l, l_t)
    if *intersects* **then**
      $V_c \leftarrow$ append intersection point p
      $A_{V_c} \leftarrow$ calculatePolygonArea(V_c)
      $A_{V_c}^{in} \leftarrow$ calculatePolygonIntersectionArea(V_c, R_t)
      $A_{V_c}^{out} \leftarrow A_{V_c} - A_{V_c}^{in}$
      $A_V^{in} \leftarrow A_V^{in} + A_{V_c}^{in}$
      $A_V^{out} \leftarrow A_V^{out} + A_{V_c}^{out}$
      $V_c \leftarrow$ clear
      $V_c \leftarrow$ append intersection point p
      $V_c \leftarrow$ append next vertex $v+1$
    else
      $V_c \leftarrow$ append current vertex v

an outlier.

4 User Study

To investigate the potential of our proposed visualization techniques for visually guiding the view of HMD users towards out-of-view objects, we designed a user study that reflects important visual search and navigation scenarios in Mixed Reality environments. Following an interactive tutorial, participants performed three different object collection tasks, during which completion time and the head rotation trajectories were recorded. Mental effort levels and overall usability scores were measured for all six visualizations in post-experiment surveys.

4.1 Subjects

A total of twenty-four subjects aged from 22 to 47 (19 male and 5 female) with a mean age of 26.63±7.05 years were recruited for the experiment. All of them were unpaid volunteers and provided their informed consent to the experiment protocol. A prerequisite for participation was that all subjects had proper color vision, as differently colored virtual objects had to be distinguished quickly. All participants passed the screening using the standard Ishihara color vision deficiency test [21], consisting of various color plates which were shown to the subjects on a smartphone application.

4.2 Task & Procedure

The experiment was conducted in an open office space. A Microsoft HoloLens HMD was used by the participants to view the MR environment. For collecting virtual objects, participants were asked to use the HoloLens clicker as input device, which proved to be more robust and novice-friendly than the tap gesture in a pilot study.

4.2.1 Calibration & Interactive Tutorial

At the beginning of the session, each participant went through the calibration application that is pre-installed on the Microsoft HoloLens. The application calibrates the display of the HMD according to the users' individual inter-pupillary distance. By going through the calibration, all users were familiarized with the basic controls of the HoloLens. Subsequently, every participant went through a tutorial application, which introduced the different visualization techniques. For each method, the tutorial offered text instructions, a visual demonstration, and the collection of three demo objects. During the tutorials, the participants had no time constraints and no measurements were taken.

4.2.2 Object Collection Scenarios

After successful completion of the tutorial, the main part of the user study followed. During three different scenarios, participants were asked to use the six visualization techniques for the task of collecting a total of eight virtual objects located in the space around them as fast as possible. Prior to each visualization technique, a start button and the name of the current technique were displayed. The experiment as well as the measurement of completion times and head rotation trajectories began as soon as the user clicked on the start button. The order of object collection scenarios was identical for every participant, starting from the simplest one and increasing in difficulty. After collecting all objects with all six techniques in one scenario, the application moved on to the next scenario.

The first scenario was a *Sequential* object collection where participants had to collect the eight objects one-by-one in sequential order. Only one object was displayed at a time and the visualization technique only showed hints for this object. After collecting an object, the next one appeared until all eight objects were collected. This scenario resembles use cases where the user is looking for a specific virtual object and is able to pass this information on to the application. This way, it is possible to limit the visualization technique to only display hints for the desired object.

In the second scenario (*Random)*, the eight objects could be collected in an arbitrary, user-determined order. All the objects were displayed at the same time and the visualization technique showed hints for all of them. This scenario is inspired by a more exploratory search task, where the user tries to quickly understand the MR environment and where all objects are potentially interesting for the user.

The third and last scenario (*Specific*) required the user to collect the virtual objects one after the other in a specific order. All eight objects were displayed at the same time and the visualization technique displayed hints for all objects. The user could, however, only collect one specific object, for which a visual hint in the form of a text message was display on the screen (e.g. "Collect the yellow sphere!"). All other objects did not react when the user tried to collect them. Once the user collected the correct object, the next hint for the next object was displayed on screen until all objects were collected. This scenario resembles a situation where the user is looking for something specific in a potentially cluttered environment, but does not pass the information to the application. Therefore, it is not possible to limit the number of hints that are displayed. We artificially create the incentive to search for a specific object by requiring the user to collect this object.

4.2.3 Post Experiment Surveys

After finishing all scenarios, participants were asked to fill out a survey. We collected data in form of their demographic information, experience with virtual and augmented reality and their experience with gaming in general. The subjects continued on to report the mental effort experienced during the different visualization techniques based on the 9-point scale of Paas [32]. Eventually, users had to fill out a System Usability Scale (SUS) survey resulting in a SUS score for every visualization technique, with a SUS score above 68 considered as above average, and a score above 80.3 considered in the top 10th percentile. The total duration of the experiment, including tutorial and post-experiment surveys, was between 30 and 50 minutes for each participant.

4.3 Object Placement

The objects that participants had to collect were positioned in a pseudo-random way around each individual user. We created a cube at the position of the users' head, which was subdivided into $3 \times 3 \times 3$ inner cubes. We then randomly positioned our virtual objects at the centers of these 27 inner cubes. Objects could not be placed into cubes directly

inside, below, above, and in front of the users, resulting in 23 potential object positions. In every scene, we placed eight objects with different shapes and uniquely identifiable colors. All objects followed the translational movements of the user, such that it was not possible to move towards or away from an object.

4.4 Design

There were two independent variables, which were controlled during the user study. The first variable had three levels and corresponded to the object collection scenario, i.e. the order, in which virtual objects had to be collected (sequential, random, or specific). We deliberately chose a fixed, difficulty-increasing ordering of object collection scenarios (i.e. 1) *Sequential*; 2) *Random*; 3) *Specific*) to keep the user study as simple as possible for participants. With the second independent variable, we controlled the visualization technique which was used to convey the 3D position of out-of-view objects. Consequently, our user study had a 3×6 within-subjects design. A 6×6 balanced Latin square matrix was employed during both the tutorial and the three object collection scenarios for randomizing the experimental conditions across study participants and object collection scenarios [44]. The two dependent variables we measured during the experiment were task completion time and the head rotation trajectory. The former defined the time necessary to collect all virtual objects for a given visualization method and collection scenario. The head rotation trajectory, expressed in spherical coordinates, corresponds to the path from the initial head rotation of a participant to the last target object and was recorded as the rotational component of the HoloLens' 3D pose at isochronal intervals.

4.5 Hypotheses

With our experimental setup in mind, we formulated the following hypothesis which was subject to an extensive statistical evaluation:

H 1. Mean task completion times with our newly introduced techniques are lower or equal to existing state of the art techniques.

Furthermore, we expected that the different visualization techniques can be classified according to the object targeting approaches employed by participants. Especially for the 3D Radar, we predicted the majority of head rotation trajectories to follow an indirect, two-way targeting approach.

5 Results

In this section, we provide a detailed analysis of the results obtained during the course of our user study. Overall, the best performing visualizations were EyeSee360 and our proposed 3D Radar, both in terms of completion times and overall usability. Furthermore, our results clearly demonstrate that participants predominantly chose the indirect, two-way object targeting approach for the 3D Radar. For all other visualizations, especially for EyeSee360, the direct, one-way targeting approach was predominantly used.

5.1 Completion Times

For comparing the time necessary to collect all virtual objects in the three different scenarios, we employed a two-way analysis of variances (ANOVA) with repeated measures ($\alpha = 0.05$) in conjunction with pairwise, Bonferroni-corrected post-hoc tests to reveal significant differences between the six visualization techniques across all three scenarios. The results are summarized in Figure 3.

There was a strong significant main effect for the object collection scenario ($F_{2,46} = 70.79, p < 0.001, \eta^2 = 0.76$). Participants were able to achieve significantly lower task completion times in the *Random* scenario compared to *Sequential* ($p < 0.001$) and *Specific* ($p < 0.001$). Furthermore, task completion time was significantly lower in the *Sequential* scenario compared to the *Specific* scenario ($p < 0.001$). A statistically significant main effect could also be recorded for the visual guidance technique ($F_{5,115} = 19.44, p < 0.001, \eta^2 = 0.46$). Mean completion times of AroundPlot ($p < 0.01$), 3D Arrows ($p = 0.001$), 3D Radar ($p < 0.001$), and EyeSee360 ($p < 0.001$) were all significantly lower than sidebARs. Similarly, completion times for the former

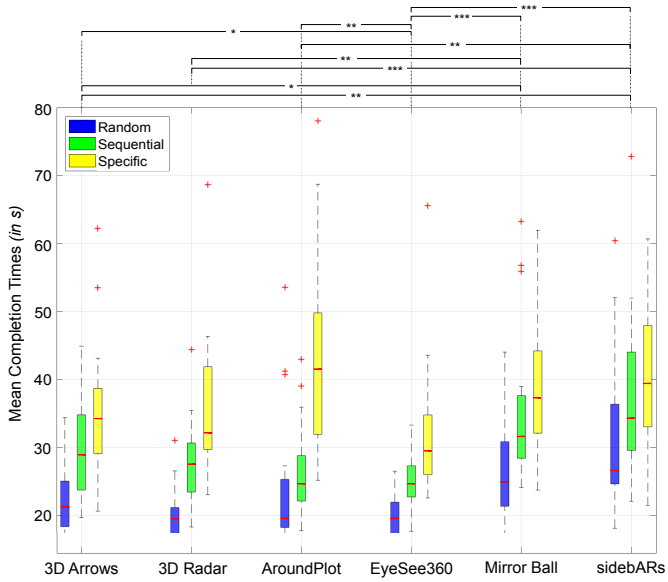

Fig. 3. Combined mean completion time results for the object collection study. In three different scenarios (sequential, random, specific), participants were asked to collect virtual shapes as fast as possible using the six different visualization techniques. Significant time differences are indicated as * ($p < 0.05$), ** ($p < 0.01$), and *** ($p < 0.001$).

three were also significantly lower than for the Mirror Ball (3D Arrows: ($p = 0.02$), 3D Radar: ($p < 0.01$), EyeSee360: ($p < 0.001$)). In addition to that, participants achieved significantly lower completion times with EyeSee360 compared to 3D Arrows ($p = 0.03$) and AroundPlot ($p < 0.001$). Interestingly, there was also a significant interaction effect between the object collection scenario and visualization technique on task completion time, such that the effect of the visual guidance technique varied depending on the type of object collection scenario ($F_{10,230} = 3.03, p = 0.001, \eta^2 = 0.12$).

In all three object collection scenarios, participants achieved the lowest mean completion times using the EyeSee360 visual guidance technique, closely followed by the 3D Radar. Both sidebARs and the Mirror Ball visualization resulted in uniformly slow task completion times in all three scenarios. While AroundPlot achieved low completion times for the *Sequential* object collection scenario, the technique came in last for the *Specific* scenario. Table 1 summarizes all completion times for each visualization and scenario.

5.1.1 Augmented Reality and Gaming Experience

In order to evaluate whether previous experiences with AR and gaming had any effect on the completion times, we performed a median split on the reported experience levels and divided participants into two groups: one for *low* and one for *high* experience with AR and gaming respectively. Both experience levels E were reported in the post-experiment survey and followed a seven-level Likert scale. For AR experience, the two groups had mean experiences $E_{AR}^{High} = 4.58 \pm 1.00$ and $E_{AR}^{Low} = 1.25 \pm 0.45$, while for gaming the mean experiences were $E_{Gaming}^{High} = 6.6 \pm 0.49$ and $E_{Gaming}^{Low} = 3.17 \pm 0.83$. In both groups with high experience levels, the mean completion times were slightly faster for each visualization technique and across all three scenarios. However, for none of these techniques the differences were statistically significant.

5.2 Head Rotation Trajectory Analysis

For analyzing how people targeted the virtual objects with the different visualization methods, we employed our previously discussed classification algorithm. As object targeting trajectories could vary greatly in the random scenario and an optimal path for collecting all objects

Table 1. Comparison of the mean completion times for all three scenarios of the object collection study, as well as average SUS-scores and cognitive load results for all six visualization techniques.

Conditions	Completion Time						SUS-Score		Cognitive Load	
	Random		Sequential		Specific					
	Mean μ	SD σ	Mean μ	SD σ	Mean μ	SD σ	Mean μ	SD σ	Mean μ	SD σ
3D Arrows	21.86s	4.68s	29.30s	7.04s	34.81s	9.15s	71.29	17.34	3.87	1.96
3D Radar	19.85s	3.87s	27.76s	5.94s	35.69s	9.82s	83.63	17.10	2.71	1.74
AroundPlot	23.17s	9.40s	26.40s	6.23s	43.03s	13.45s	79.19	12.82	3.13	1.23
EyeSee360	18.58s	3.63s	24.76s	4.04s	31.84s	9.38s	89.27	10.31	1.55	0.68
Mirror Ball	26.61s	9.38s	37.23s	7.40s	39.88s	15.29s	44.35	19.87	5.45	2.11
sidebARs	31.32s	10.01s	38.66s	10.76s	40.91s	14.95s	51.05	19.67	5.26	2.35

is difficult to define in this scenario, we restricted our analysis of head rotation data to the other two scenarios with sequential and specific object collection orders. In Figure 4, all 48 targeting trajectories (8 target objects $\times$ 6 visualization techniques) of one particular participant for the sequential scenario are depicted.

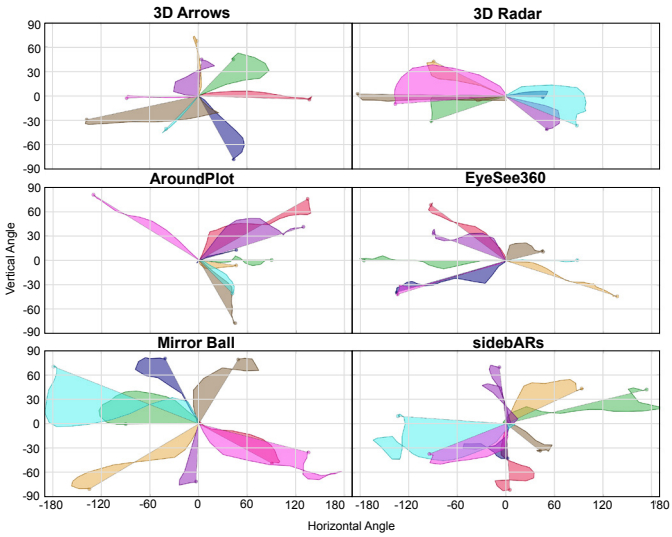

Fig. 4. All trajectories from one participant for the six visualization techniques during the sequential collection scenario. The different colors correspond to the colors of the virtual objects that were collected with the respective trajectory. Note: No distinction between A_V^{in} and A_V^{out} is made.

5.2.1 Outlier Removal

Prior to classification, outlier trajectories were identified and removed from the data. There are two types of outliers: the first one is characterized by a trajectory length that exceeds the length of l_t (i.e. the line from the origin to the target point t) by a certain percentage. For our analysis, this percentage was set to 300, such that all trajectories whose length was 3-times longer than that of l_t were removed. Furthermore, we classified all those trajectories as outliers, where the target virtual object was close to one the axes $x = 0$ or $y = 0$. In these cases, it is not possible to distinguish between a one-way approach and a two-way approach. For our analysis, this threshold was set to 10° for both the horizontal and vertical angle. For the sequential scenario, the percentages of trajectories classified as outliers using these two criteria were 19.26%, distributed in the following way: 3D Arrows (3.04%), 3D Radar (2.60%), AroundPlot (2.69%), EyeSee360 (2.86%), Mirror Ball (4.51%), and sidebARs (3.56%). In the specific scenario, the percentage of outliers was almost the same with 19.10% of all trajectories: 3D Arrows (3.56%), 3D Radar (3.13%), AroundPlot (2.95%), EyeSee360 (1.91%), Mirror Ball (3.47%), and sidebARs (4.08%).

5.2.2 Class Definitions & Thresholds

For all remaining trajectories (*inliers*), Algorithm 1 is applied to calculate the areas A_V^{in} and A_V^{out}. Based on these two metrics, the trajectories are classified as $T_{OneWay}, T_{TwoWay}, T_{InBetween}$, or $T_{Unclassifiable}$, according to equation 1. As described in section 3.3, the classes are used to describe which object targeting approach was used by the participant. For the thresholds α and β in equation 1, we set the values to 20% and 30% respectively to have a rather conservative estimate and allow for moderate marginals. Figure 5 illustrates the distribution of $(\frac{A_V^{in}}{A_{R_t}}, \frac{A_V^{out}}{A_{R_t}})$ ratio pairs as well as the classification boundaries defined by the two thresholds α and β for both the sequential and specific scenario.

5.2.3 Sequential Scenario

After the removal of outliers, there was still a moderate percentage of trajectories that could not be classified due to large portions of the area contributing to A_V^{out}. Interestingly, the visualizations with the smallest percentages of unclassifiable trajectories were also the ones with the lowest mean completion times in the sequential scenario, namely EyeSee360 (11.61%), AroundPlot (15.93%), and 3D Radar (16.67%). The Mirror Ball had the largest percentage of trajectories classified as $T_{Unclassifiable}$ with 28.16%.

The largest percentages of trajectories for all six visualizations was classified as $T_{InBetween}$. For these trajectories, the relative percentage of A_V^{in} was in the range of $[\alpha, 100 - \alpha]$ and thus could not be assigned clearly to one of the two targeting approaches, cf. Figure 5.

From the remaining trajectories, between $30\% - 40\%$ were classified as either T_{OneWay} or T_{TwoWay}, which is still an acceptable percentage considering the large inter-operator variabilities of targeting virtual objects in AR. Our results demonstrate that participants predominantly chose a one-way targeting approach for all visualization techniques, with the exception of the 3D Radar. This was especially evident for the EyeSee360 method, where only 2.68% of all trajectories were classified as T_{TwoWay}, compared to 34.82% classified as T_{OneWay}. The 3D Radar was the only method which favored a two-way approach, with 26.47% of all trajectories classified as T_{TwoWay}, compared to 10.78% for T_{OneWay}. Table 2 provides the exact percentages for all six visualization techniques.

5.2.4 Specific Scenario

In the specific collection scenario, we observed the same overall trends, reaffirming our results from the sequential scenario. Participants pre-

Table 2. Results of the head rotation trajectory classification. For both the sequential and specific collection scenario, the percentages of all four classes indicating different object targeting approaches are summarized individually for all six visualization techniques.

Conditions	Sequential				Specific			
	One-Way	Two-Way	In-Between	Unclassifiable	One-Way	Two-Way	In-Between	Unclassifiable
3D Arrows	25.69%	8.26%	41.28%	24.77%	26.55%	10.62%	43.36%	19.47%
3D Radar	10.78%	26.47%	46.08%	16.67%	9.73%	24.77%	46.28%	19.20%
AroundPlot	21.24%	11.50%	51.33%	15.93%	32.08%	20.75%	33.96%	13.21%
EyeSee360	34.82%	2.68%	50.89%	11.61%	40.37%	7.34%	41.28%	11.01%
Mirror Ball	29.13%	12.62%	30.10%	28.16%	26.61%	11.93%	32.11%	29.36%
sidebARs	21.37%	17.95%	36.75%	23.93%	21.90%	14.29%	42.86%	20.95%

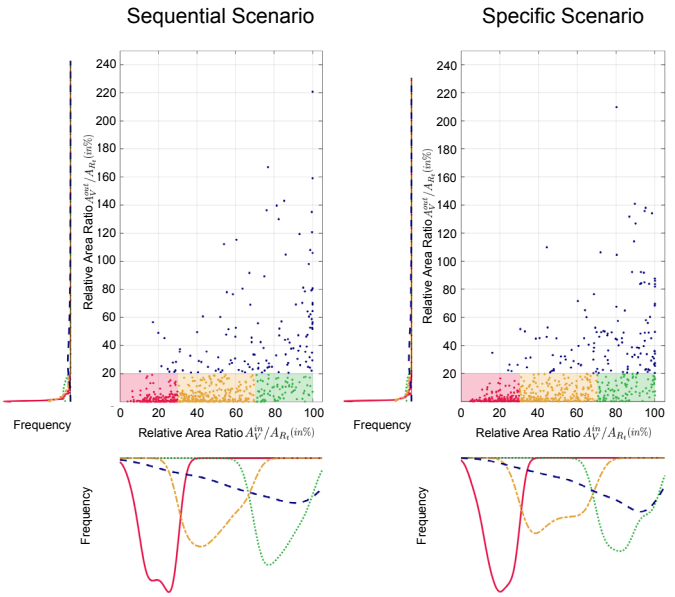

Fig. 5. Distribution of head rotation trajectories for the sequential and specific collection scenario without outliers. Each point corresponds to one trajectory, which is described in terms of the two ratios A_V^{in}/A_{R_t} and A_V^{out}/A_{R_t}. The classification boundaries are indicated by three colored rectangles: T_{OneWay} (red), $T_{InBetween}$ (yellow), T_{TwoWay} (green). All other trajectories (blue) are assigned to $T_{Unclassifiable}$.

dominantly used the two-way targeting approach with the 3D Radar visualization (24.77% compared to 9.73% for T_{OneWay}). For all other techniques, the one-way targeting approach was clearly preferred by the majority, again most notably for EyeSee360 ($T_{OneWay} = 40.37\%$, $T_{TwoWay} = 7.34\%$). Surprisingly, the overall percentages of trajectories that could not be classified was slightly lower compared to the (theoretically easier) sequential collection scenario.

5.2.5 System Usability Scale

The perceived usability of all six visualization techniques was studied by means of a SUS questionnaire, cf. Table 1. Very high scores were recorded for both EyeSee360 (89.27±10.31) and the 3D Radar (83.63±17.10), followed by AroundPlot (79.19±12.82) and 3D Arrows (71.29±17.34). The Mirror Ball and sidebARs were clearly outperformed and received much lower SUS-scores with 44.45±19.87 and 51.05±19.67 respectively. A repeated measures ANOVA revealed a significant effect of visualization technique on SUS total score ($F_{5,115} = 27.12, p < 0.001, \eta^2 = 0.54$). Post-hoc tests with Bonferroni correction showed that 3D Arrows, 3D Radar, AroundPlot, and EyeSee360 had significantly higher usability ratings over the Mirror Ball ($p < 0.001$ for all four techniques) and sidebARs (3D Arrows: $p = 0.02$, for all four techniques: $p < 0.001$). Furthermore, EyeSee360 achieved a significantly higher SUS score over 3D Arrows ($p = 0.002$).

5.2.6 Mental Effort

For comparing the mental effort experienced by the subjects during the three different collection scenarios, non-parametric Kruskal-Wallis tests were employed. They showed a significant effect of the visualization technique on the experienced mental effort ($H(5) = 60.06, p < 0.001, \eta^2 = 0.40$). Post-hoc tests revealed significantly lower mental effort levels for EyeSee360 compared to 3D Arrows ($p < 0.001$), AroundPlot ($p = 0.007$), the Mirror Ball ($p < 0.001$), and sidebARs ($p < 0.001$). Similarly, the experienced mental work levels were significantly lower for the 3D Radar compared to the Mirror Ball ($p < 0.001$) and sidebARs ($p < 0.001$). Table 1 provides the mean mental effort scores for all visualization techniques, which were reported according to a 9-point scale of Paas [32].

6 Discussion

Three main observations may be inferred from the results of our user study. First, the performance of our proposed 3D Radar is on par with state-of-the-art techniques for visualizing out-of-view objects. Even if not outperforming EyeSee360, our results indicate that the 3D Radar provides an intuitive way for quickly navigating MR environments. In contrast to other techniques, the 3D Radar offers the potential to encode a variety of information in a small portion of the users' view, including both the direction towards a target object as well as its distance. By varying the number and radii of the concentric circles, the 3D Radar additionally allows for encoding various ranges of distances, which is not possible with techniques such as EyeSee360 or 3D Arrows. This way, the inner circles can be used to convey information about objects in close proximity, while the outer circles display objects that are possibly far away from the user. In our study, we used up- and downward facing triangles as proxy icons to indicate whether an object is above or below the user. However, this information is also encoded in the color of both the projection circle and the line connecting the proxies and the projections. Therefore, it is possible for other applications to replace the triangles with small pictures of the individual objects to gain an even better understanding of the environment. While the proxies in Aroundplot and EyeSee360 could also be replaced by more specific icons, the 3D Arrows can hardly support this type of encoding. Additionally, the 3D Radar takes up significantly less screen estate than the slightly superior EyeSee360 method. For certain applications, that continuously require an unobstructed view onto the real scene, the 3D Radar might be the preferred choice. These potential advantages need to be further investigated in future experiments.

The second observation concerns the inferior performance of the Mirror Ball, both in terms of collection times of virtual objects and overall usability. We think there are several factors that contributed

to these results: first of all, interactions with spherical mirrors hardly occur in our everyday life. Therefore, interpreting the reflections on the Mirror Ball is not intuitive and requires high mental effort. This is reflected in the highest cognitive load scores during our questionnaires for the Mirror Ball. Secondly, inferring the location of objects by looking at their reflections on the Mirror Ball is complicated by the fact, that the physical environment is not reflected on the Mirror Ball. This could potentially be achieved by having either a dynamic 3D model of the environment or a 360° video from the users' perspective. However, the resulting visualization would no longer correspond to a visual attention guide towards virtual objects, but correspond to a an alternative, distorted representation of the world. We attempted to improve the perception of the Mirror Ball by reflecting a skybox in form of a spherical grid, which only partially fulfilled its purpose. Additionally, users reported that it is difficult to recognize whether an object is behind or in front of the user, because the users' body is not reflected by the Mirror Ball. This issue could potentially be solved by placing a semi-transparent representation of the users' body in the virtual world. A technical issue that was specifically evident for the Mirror Ball was screen tearing due to the limited computing power of the HoloLens. This could possibly be improved by employing a custom shader that handles reflections more efficiently or representing virtual objects in form of their projections on the Mirror Ball rather than reflections. Considering the first two observations, we can only accept our hypothesis H1 for the 3D Radar.

A third important observation was the tendency of participants to favor indirect, two-way targeting trajectories when using the 3D Radar, which confirmed our initial assumption. The comparable completion time results between the 3D Radar and EyeSee360 are especially interesting in the light of this observations, as users generally followed longer trajectories with the former. Our area-based trajectory classification algorithm proved to be a suitable tool for discriminating between the different object targeting approaches. On the other hand, the largest percentage of trajectories was still classified as a mixture between a one-way and two-way targeting approach for all visual guidance techniques, indicating that neither of the two is clearly superior to the other. From an ergonomic point of view, an *optimal* head trajectory towards a target object in MR environments might be a much more complex function than the two we focused on during our user study. Humans have been shown to perform physical tasks in ways that minimize not only path length, but also effort and discomfort. These concepts have been applied before in the area of robotics to increase human-like behavior of anthropomorphic robots [24, 25]. While our analysis of the head rotation data presents a first step towards better understanding the rotational movement patterns in MR environments, further investigation is required. Introducing a biomechanical model of the head & neck physiology into the trajectory analysis could be used to better understand which parameters to optimize for and eventually to implement visual guidance techniques that favor such physically-motivated trajectories [27]. Furthermore, a common phenomenon which frequently occurred for all types of visualizations were *overshoots*, i.e. moving too far in the direction of the target objects with a subsequent reversal. Similarly, directional corrections where users initially moved to a wrong direction could be observed especially at the beginning of head trajectories, but were not included in our analysis. Investigating the trajectory data with respect to these two patterns and developing mitigation strategies will be an interesting area for future research. Another potential direction for future research is the integration of eye tracking. Examining how often and for how long users focus on a certain visualization could provide further insights into the perceptual aspects of visualizing out-of-view objects.

6.1 Study Limitations

In terms of study limitations, we did not include a baseline condition, i.e. a *no-visualization* condition, during our user study to measure the exact quantitative benefits of the six different visualization techniques compared to not having any visual guidance. However, this is in accordance with previously published related work (especially with all four papers that introduced the techniques that served as comparison during our user study) and we argue that a *no-visualization* condition resembles a completely random search rather than an exploratory or goal-oriented search which is enabled by visual guidance techniques.

Additionally, our user study was limited to investigating the effects of different visual guidance techniques in the case of *static* virtual objects. For *dynamically* moving objects, the visualization needs to continuously convey information about an objects' position and potentially about the expected future trajectory to the user. An interesting work was recently published by Walker et al. which investigates the communication of a robots' motion intent using AR during human-robot-interaction [41]. We believe that especially our 3D Radar technique would be well suited for both visualizing expected trajectories as well as tracking a moving object due to its capabilities of displaying absolute object positions. This is inherently difficult for all other visual guidance techniques, as they primarily encode relative object positions. However, future studies are necessary to investigate the effects of visual guidance techniques in the case of dynamically moving objects.

7 Conclusion

In this paper, we presented the Mirror Ball and the 3D Radar as two novel approaches for visualizing out-of-view objects in Mixed Reality environments. We evaluated their potential to convey information about the location of virtual objects in a user study which reflected three scenarios of real-world exploratory and goal-oriented search tasks. In our experiments, performance for the 3D Radar was comparable to the best state-of-the-art techniques, while at the same time offering several advantages such as smaller field-of-view coverage and additional information encoding potential. For the Mirror Ball, several perceptual challenges have to be further investigated before comparable results can be expected. In addition to the two novel visualization approaches, we proposed a new area-based method for analyzing and classifying head rotation trajectories. We demonstrated that different visualizations encourage distinct object targeting approaches in MR. We hope that this work opens the path for further research in the area of head rotation analysis in MR environments in order to deeply understand the underlying perceptual effects of different out-of-view object visualization techniques.

References

[1] https://en.wikipedia.org/wiki/Elite:_Dangerous.

[2] https://en.wikipedia.org/wiki/Eve:_Valkyrie.

[3] https://en.wikipedia.org/wiki/Star_Citizen.

[4] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.

[5] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P. S. Yu. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 34–43. ACM, 2006.

[6] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE transactions on Image Processing*, 16(7):1912–1919, 2007.

[7] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Transactions on Multimedia*, 9(1):58–65, 2007.

[8] P. Baudisch and R. Rosenholtz. Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pp. 481–488. ACM, New York, NY, USA, 2003.

[9] J. Bian, D. Tian, Y. Tang, and D. Tao. A survey on trajectory clustering analysis. *arXiv preprint arXiv:1802.06971*, 2018.

[10] F. Biocca, A. Tang, C. Owen, and F. Xiao. Attention funnel: Omnidirectional 3d cursor for mobile augmented reality platforms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pp. 1115–1122. ACM, New York, NY, USA, 2006.

[11] S. Burigat and L. Chittaro. Navigation in 3d virtual environments: Effects of user experience and location-pointing navigation aids. *International Journal of Human-Computer Studies*, 65(11):945–958, 2007.

[12] S. Burigat, L. Chittaro, and S. Gabrielli. Visualizing locations of off-screen objects on mobile devices: A comparative evaluation of three approaches.

In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '06, pp. 239–246, 2006.

[13] M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pp. 61–70. ACM, New York, NY, USA, 2001.

[14] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 792–803. VLDB Endowment, 2004.

[15] S. R. Ellis, B. D. Adelstein, and K. Yeom. Human control in rotated frames: anisotropies in the misalignment disturbance function of pitch, roll, and yaw. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, pp. 1336–1340. SAGE Publications Sage CA: Los Angeles, CA, 2012.

[16] D. R. Faria and J. Dias. 3d hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1284–1289. IEEE, 2009.

[17] U. Gruenefeld, D. Ennenga, A. E. Ali, W. Heuten, and S. Boll. Eyesee360: Designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, pp. 109–118, 2017.

[18] U. Gruenefeld, D. Hsiao, W. Heuten, and S. Boll. Eyesee: beyond reality with microsoft hololens. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pp. 148–148. ACM, 2017.

[19] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani. Wedge: Clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 787–796. ACM, New York, NY, USA, 2008.

[20] S. G. Gustafson and P. P. Irani. Comparing visualizations for tracking off-screen moving targets. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pp. 2399–2404. ACM, New York, NY, USA, 2007.

[21] S. Ishihara. Series of plates designed as tests for colour-blindness. 1936.

[22] H. Jo, S. Hwang, H. Park, and J. hee Ryu. Aroundplot: Focus+context interface for off-screen objects in 3d environments. *Computers & Graphics*, 35(4):841 – 853, 2011. Semantic 3D Media and Content.

[23] R. Jonker, G. De Leve, J. Van Der Velde, and A. Volgenant. Rounding symmetric traveling salesman problems with an asymmetric assignment problem. *Operations Research*, 28(3-part-i):623–627, 1980.

[24] O. Khatib, L. Sentis, J. Park, and J. Warren. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(01):29–43, 2004.

[25] O. Khatib, J. Warren, V. De Sapio, and L. Sentis. Human-like motion from physiologically-based potential energies. In *On advances in robot kinematics*, pp. 145–154. Springer, 2004.

[26] J.-G. Lee, J. Han, X. Li, and H. Gonzalez. Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment*, 1(1):1081–1094, 2008.

[27] S.-H. Lee and D. Terzopoulos. Heads up!: biomechanical modeling and neuromuscular control of the neck. *ACM Transactions on Graphics (TOG)*, 25(3):1188–1198, 2006.

[28] S. Matsuzoe, S. Jiang, M. Ueki, and K. Okabayashi. Intuitive visualization method for locating off-screen objects inspired by motion perception in peripheral vision. In *Proceedings of the 8th Augmented Human International Conference*, p. 29. ACM, 2017.

[29] J. McCrae, M. Glueck, T. Grossman, A. Khan, and K. Singh. Exploring the design space of multiscale 3d orientation. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pp. 81–88. ACM, New York, NY, USA, 2010.

[30] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[31] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):623–635, 1980.

[32] F. G. Paas. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of educational psychology*, 84(4):429, 1992.

[33] C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Softw.*, 12(2):21–32, Mar. 1995.

[34] P. Renner and T. Pfeiffer. Attention Guiding Techniques using Peripheral Vision and Eye Tracking for Feedback in Augmented-Reality-based Assistance Systems. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2017.

[35] T. Schinke, N. Henze, and S. Boll. Visualization of off-screen objects in mobile augmented reality. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pp. 313–316. ACM, New York, NY, USA, 2010.

[36] T. Siu and V. Herskovic. Sidebars: Improving awareness of off-screen elements in mobile augmented reality. In *Proceedings of the 2013 Chilean Conference on Human - Computer Interaction*, ChileCHI '13, pp. 36–41. ACM, New York, NY, USA, 2013.

[37] 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. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 331–340. ACM, New York, NY, USA, 2014.

[38] I. E. Sutherland and G. W. Hodgman. Reentrant polygon clipping. *Communications of the ACM*, 17(1):32–42, 1974.

[39] M. Trapp, L. Schneider, C. Lehmann, N. Holz, and J. Dllner. Strategies for visualising 3d points-of-interest on mobile devices. *Journal of Location Based Services*, 5(2):79–99, 2011.

[40] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pp. 673–684. IEEE, 2002.

[41] M. Walker, H. Hedayati, J. Lee, and D. Szafir. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 316–324. ACM, 2018.

[42] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou. An effectiveness study on trajectory similarity measures. In *Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137*, pp. 13–22. Australian Computer Society, Inc., 2013.

[43] K. Weiler and P. Atherton. Hidden surface removal using polygon area sorting. In *ACM SIGGRAPH computer graphics*, vol. 11, pp. 214–222. ACM, 1977.

[44] E. Williams. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Chemistry*, 2(2):149–168, 1949.

[45] P. T. Zellweger, J. D. Mackinlay, L. Good, M. Stefik, and P. Baudisch. City lights: Contextual views in minimal space. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 838–839. ACM, New York, NY, USA, 2003.