EyeSee360: Designing a Visualization Technique for Out-of-view Objects in Head-mounted Augmented Reality

Uwe Gruenefeld
Oldenburg, Germany
University of Oldenburg
uwe.gruenefeld@uol.de

Dag Ennenga
Oldenburg, Germany
University of Oldenburg
dag.ennenga@uol.de

Abdallah El Ali
Oldenburg, Germany
University of Oldenburg
abdallah.el.ali@uol.de

Wilko Heuten
Oldenburg, Germany
OFFIS - Institute for IT
wilko.heuten@offis.de

Susanne Boll
Oldenburg, Germany
University of Oldenburg
susanne.boll@uol.de

ABSTRACT

Head-mounted displays allow user to augment reality or dive into a virtual one. However, these 3D spaces often come with problems due to objects that may be out of view. Visualizing these out-of-view objects is useful under certain scenarios, such as situation monitoring during ship docking. To address this, we designed a lo-fi prototype of our EyeSee360 system, and based on user feedback, subsequently implemented EyeSee360. We evaluate our technique against well-known 2D off-screen object visualization techniques (Arrow, Halo, Wedge) adapted for head-mounted Augmented Reality, and found that EyeSee360 results in lowest error for direction estimation of out-of-view objects. Based on our findings, we outline the limitations of our approach and discuss the usefulness of our developed lo-fi prototyping tool.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**; **Information visualization**; *User studies*;

KEYWORDS

Head-mounted; augmented reality; out-of-view; off-screen; visualization techniques; peripheral awareness

ACM Reference format:
Uwe Gruenefeld, Dag Ennenga, Abdallah El Ali, Wilko Heuten, and Susanne Boll. 2017. EyeSee360: Designing a Visualization Technique for Out-of-view Objects in Head-mounted Augmented Reality. In *Proceedings of SUI '17, Brighton, United Kingdom, October 16–17, 2017,* 10 pages.
DOI: 10.1145/3131277.3132175

978-1-4503-5486-8/17/10…$15.00
DOI: 10.1145/3131277.3132175

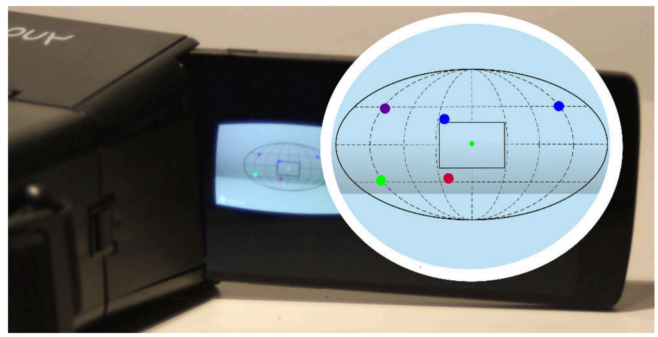


Figure 1: Google Cardboard with zoom-in on EyeSee360 visualization. ***Best seen in color.***

1 INTRODUCTION

In recent years, Augmented Reality (AR) and Virtual Reality (VR) technology have experienced a sustained upswing (e.g., for navigation [21], interaction [24] or collaboration [18]). The fundamental idea of these technologies is to alternate the percieved reality by augmenting or virtualizing it. Experienced in a head-mounted device, users can use such technologies hands-free and while mobile. This has advantages in many spatial working environments where machines have to be operated by hand (e.g. emergency rooms [27]) or in situations where the user is moving. However, due to biological factors, the human visual range is restricted, whereby humans can only perceive parts of their environment at once while everything else is hidden out of view. This problem is amplified when a head-mounted device is further limiting the field of view (e.g., a Virtual Reality headset). Essentially, this is a problem that arises in situations where spatially distributed objects that are out of view need to be observed or tracked. In previous work [8], a first solution for perceiving information about out-of-view objects has been suggested for the AR space. However, the proposed solution encodes only direction information limited by a decreasing accuracy for increasing direction angles between the users line of sight and out-of-view objects. In other words, the problem of perceiving information about out-of-view objects in the 360 degrees around the user has not yet been solved for the AR or VR space, especially given the range of head-mounted devices today.

In this paper, we investigate the problem of how to represent out-of-view objects in head-mounted Augmented Reality. To do so, we visualize the direction and distance towards out-of-view objects with representations of them on a head-mounted device. Here, we came up with a new visualization technique for out-of-view objects. We compared it to adaptations of well-known 2D techniques (Arrow, Halo and Wedge) proposed in [8]. These adaptions are limited to visualizing the direction towards out-of-view objects and therefore, we can only evaluate our direction encoding with it. To our knowledge, there is no previous work that addresses visualization of direction and distance towards out-of-view objects at the same time.

Our goals in this paper are to design, implement and evaluate visualization techniques for out-of-view objects in head-mounted Augmented Reality. We ask the following research questions: (1) What is a suitable way for visualizing out-of-view objects in EyeSee360, our system that uses the 360° space around the user? (2) What are suitable representations for encoding distance of out-of-view objects? (3) Which visualization technique (EyeSee360, Arrow, Halo, Wedge) for head-mounted Augmented Reality results in the best user performance with respect to direction estimation, workload and perceived usability?

We make two research contributions to the Spatial User Interaction community:

(1) We present a lo-fi head-mounted prototyping tool that allows quickly testing and refining design ideas.
(2) We design, implement and evaluate a novel out-of-view object visualization technique **EyeSee360** that can serve as a starting point for future work and compare it to adapted 2D off-screen visualization techniques (Arrow, Halo, Wedge).

2 RELATED WORK

Three main approaches have been proposed for the encoding of off-screen objects: Overview+detail (O+D), Contextual views, and Focus+context (F+C) [6, 9]. In the O+D approach, a miniature map of the surroundings is shown in addition to the detailed view (e.g. as a road map). The problem with this approach is the cognitive load to integrate all views into one overall understanding of the map. Contextual views and F+C both overlay the current focus with context information. The approaches differ in the kind of transition between focus and context. In the F+C approach the transition is soft (e.g. fisheye-views that convey a distorted view [25]) and for Contextual views the transition is hard (e.g. arrows pointing into off-screen space [5]). Both approaches seem to be feasible to be used for head-mounted Augmented Reality. A list of relevant off-screen visualization techniques using these two approaches is shown in Table 1. Aside from research, off-screen visualization is frequently used in computer games. An early example in 2D games is Tecmo Bowl (1987)[1] that uses simplified arrows. In 3D games like X-Wing (1993)[2] or the newer Eve: Valkyrie (2016)[3] a radar-like visualization is used. But since these commercial solutions do not offer any user-driven evaluations we consider them more as a source of inspiration for our visualization technique.

[1]https://en.wikipedia.org/wiki/Tecmo_Bowl
[2]https://de.wikipedia.org/wiki/Star_Wars:_X-Wing
[3]https://en.wikipedia.org/wiki/Eve:_Valkyrie

Technique	Objects	Dim.	HMD	Distance
City Lights [33]	multiple	2D	no	yes
Halo [1]	multiple	2D	no	yes
Arrows [5, 13]	multiple	2D	no	yes
Wedge [9]	multiple	2D	no	yes
EdgeRadar [10]	multiple	2D	no	yes
SidebARs [27]	multiple	2D	no	no
Aroundplot [14]	multiple	3D	no	no
3D-Halo Circle [32]	multiple	3D	no	no
3D-Halo Projection [32]	multiple	3D	no	yes
3D-Arrows [4, 26, 31]	multiple	3D	no	yes
Attention Funnel [2]	single	3D	yes	no
ParaFrustum [30]	single	3D	yes	no

Table 1: Comparison between prior off-screen object visualization techniques.

2D Visualization Techniques. Zellweger et al. [33] presented City Lights which provides contextual information along the borders but it is difficult to guess the position of the off-screen objects. Therefore, Halo was suggested as an improvement [1]. It uses circles drawn with their center around the off-screen object and cut the border of the screen slightly. However, a problem of Halo is cluttering, which is the accumulation of many Halos in corners. Several studies compared Halo with Arrow approaches [5, 13], where Arrows with fixed sizes performed worse while scaled arrows performed better. Also the amount of visible objects have a high impact on the performance. An improvement of Halo to avoid cluttering is Wedge [9]. Instead of circles they propose isosceles triangles, which use less space. Given the foregoing, we choose Arrow, Halo and Wedge to compare our approach against, since they are well studied and lend themselves easily to head-mounted AR views. Another approach is by Siu and Herskovic [27], who propose SidebARs for improving awareness of off-screen objects. However their AR system is not head-mounted, and objects were presented only on a 2D plane. Finally, Gustafson et al. [10] present EdgeRadar, which shows how to use focus and context for off-screen object visualization without distortion effects – their approach serves as inspiration to our EyeSee360 technique.

3D Visualization Techniques. It seems natural for a head-mounted AR system to use a strategy for off-screen objects in 3D. However, most solutions in this field have limitations making them less feasible for visualizing out-of-view objects in a head-mounted device. AroundPlot from Jo et al. [14] uses a mapping from 3D spherical coordinates to 2D orthogonal fisheye and dynamic magnification. However, an orthogonal mapping leads to the corner-density problem and the positive and negative effects of dynamic magnification need to be examined in further studies. 3D Halo Circle and 3D Halo Projection from Trapp et al. [32] have been proposed for virtual environments. Both approaches suffer from visual clutter. Moreover, other solutions like 3D-Arrows are limited to objects in front of the user (e.g., [4, 26, 31]) or to one off-screen object at a time (e.g. Attention Funnel [2]). For Attention Funnel and for ParaFrustum [30], head-mounted devices are used to display the visualization technique. A different approach to represent out-of-view objects in Virtual Reality is a "world in miniature" but it uses an O+D approach and is therefore not considered [28].

3 GENERAL APPROACH

To address the problem of out-of-view objects in 3D space, we divide the problem into two subproblems: visualizing the direction of an out-of-view object and visualizing the distance towards this object. This makes sense since many use cases require only a visualization of the direction information (e.g. monitoring tasks). As a first step, we attempt at encoding direction information. To do so, we drew on prior work on mobile (small screen estate) off-screen visualizations (see Table 1). From the various previous approaches, we identified two approaches that are fitting for our context of use.

The first approach is **Focus+context**, which uses a soft transition, and the second approach **Contextual Views**, which uses a hard transition. Both overlay the screen border with context information as these approaches are drawn from 2D displays. Only these two seem fitting for head-mounted Augmented Reality as the information overlays are not centered on-screen, and displayed peripherally along the outer part of the screen. This means the user's direct (foveal) line of sight remains unaffected by visual changes [16]. Furthermore, across both approaches the visualizations use representations of the off-screen objects that are placed in the same direction as the off-screen object itself, and thereby are in line with the human frame-of-reference [14].

As a starting point, we found that the **Focus+context** approach is more effective for developing a new visualization technique because the design of the representations is easier to perceive [19]. This is relevant since the border of the screen becomes the periphery of the user in a head-mounted device. In addition, EdgeRadar showed that moving objects can be tracked more accurately with it than with Halo[10]. This is relevant since EdgeRadar [10] served as inspiration for our technique. To also consider techniques from the **Contextual Views** approach we evaluate against Arrow, Halo and Wedge adapted for head-mounted Augmented Reality [8].

4 PART I: EYESEE360 CONCEPT VALIDATION

As a first step, we explore the design and development of our **EyeSee360** technique for visualizing out-of-view objects in the 360° surrounding the user.

4.1 Concept Definition

We created three different variants for the encoding of direction information and three different levels of support in EyeSee360. In the peripheral field of view, an inner ellipse and an outer ellipse are drawn. The inner ellipse is the representation of the field of view and is sized so as not to occlude the user's focus. Objects within the field of view are not considered out-of-view objects. The outer ellipse represents the 180° line (behind the user) from top (90°) to bottom (90°). We choose an ellipse because of human color perception. In Figure 2 it is visible that the binocular human vision is wider compared to its height. Further, we need to represent up to 180° behind the user and only up to 90° to the top.

Objects that are not currently visible are displayed as a dot between the outer and inner ellipse. Position of this dot determines its position out of view. Left and right are shown up to 180° and up and down up to 90°. This makes it possible to visualize any position of out-of-view objects in 3D space. The three different variants for direction encoding we explored are as follows: Variant 1: Only an inner and outer ellipse (Ellipses). Variant 2: Ellipses + 0° lines. Variant 3: In addition to 0° lines, there are now helplines for 45° steps (Helplines). Additionally we have three different variants for distance encoding: Variant 1: size, Variant 2: color and Variant 3: size and color; see Figure 3. The cross seen in Figure 3 will be placed in the center of the human field of view.

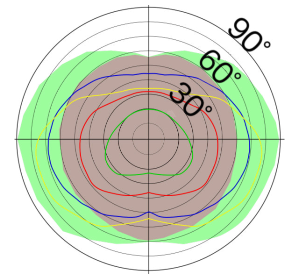
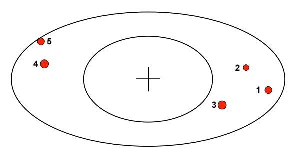
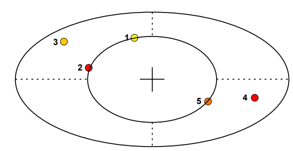


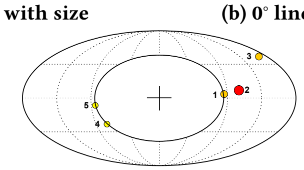
Figure 2: The perception of colors in the human field-of-view. Each colored line represents the outer border of its percieved limits [15, 29]. Light green represents the binocular field of view and dark gray represents the monocular field of view [22]. *Best seen in color.*



(a) Elipse with size



(b) 0° lines with color



(c) Helplines with combination

Figure 3: Variants of EyeSee360. *Best seen in color.*

To represent the distance of objects, the following representations can be considered: size, color, brightness and animation. Brightness was excluded from the selection early on, as it would leave objects with low alpha values to be faded. Since brightness is influenced by the environment, unintentional changes in perception can occur. An animation could let us animate objects (e.g. blinking) with a high frequency, however animation in the periphery triggers attention towards it and therefore should only be used for attention shifting and not for continuously visualization of information.

The two other forms of size and color have been selected as possible candidates for distance representation. Colors are already important when it comes to attention (e.g., traffic lights). On the other hand, we have a natural association with distance and size. The further away something is, the smaller it looks. A combination of both forms is also possible. Here, it is important that the two colors are clearly distinguishable on every step of their color gradient. Therefore, we choose red and yellow for our experiment. Red for closer objects because we consider them to be more important and yellow for objects far away. Further, it is important that the sizes of the different distance encodings are clearly distinguishable. However, a limit of the size must be considered. Very large representations of the out-of-view objects lead to more frequent overlapping. The size, color and their combination for distance representation are therefore compared (variants) in the following concept validation study.

4.2 Concept Validation Study

To validate the concept of EyeSee360, we ran a concept validation study. Our goal was to evaluate if users are able to determine the position of out-of-view objects with EyeSee360. To test our concept, we developed a rapid prototyping tool that allows researchers to rapidly test non-changing head-mounted see-through views without implementation. With this tool, we could use simple slides to get users' impression. Here, we needed to develop our own prototyping tool as previous tools such as the PapAR tool by Lauber et al. [17] do not consider head-mounted devices, or focus on the interface elements [7] and not on visualizing information. Our prototyping tool is not limiting the human field-of-view because of transparent materials.

4.3 Designing the Lo-fi Prototyping Tool

We laser cut our prototyping tool (Figure 4). We used Plexiglass manufactured glasses which allow sliding in film slides (known as transparencies or viewfoil) in the front. The user is then able to explore static see-through layers and their effects on a perceived environment. It is even possible to have more layers. Since no lenses were used, care must be taken that human perception does not allow any content to be focused directly in front of the user. Accordingly, a suitable distance between the film and the human eye must be taken into account. It should be noted that this distance must always be greater with increasing biological age.

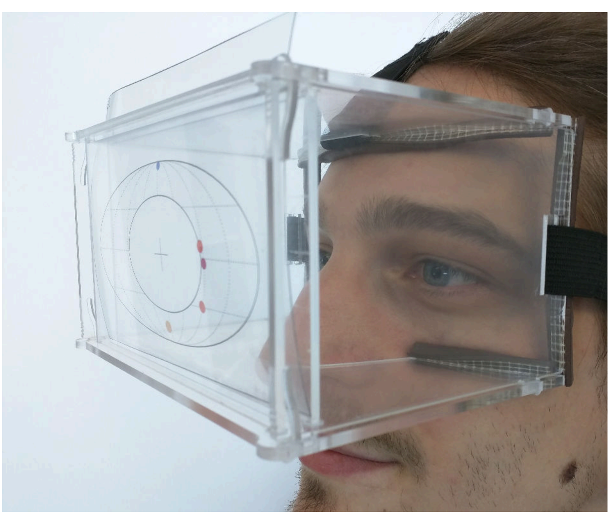


Figure 4: Person wearing our lo-fi AR prototyping tool. *Best seen in color.*

4.4 Study Design

To evaluate the performance of the three variants (color, size, combination) described above we conducted a comparative user study. The user study was designed as a lab study and took place in an empty office room with white walls and darkened windows to avoid effects of different light conditions. We lit the room with artificial light (around 600 lux). We used quantitative methods to objectively evaluate the performance combined with SUS questionnaires to gain insights into the perceived usability of these variants. For this validation study, we fixed several parameters: (1) All objects were homogenously distributed out of view in 3D space to fit all directions and distances equally (2) The users viewing angle onto the objects and the position of the objects did not change throughout the study (3) The number of displayed objects was fixed to 5. We investigated user performance (accuracy) and subjective variable perception for object direction and distance.

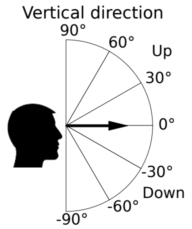
In this study, we have two independent variables: Support with 3 levels (basic vs. 0° lines vs. help-lines), and Attribute with 3 levels (size vs. color vs. combination). Since we could not separate direction from distance information, it was not necessary to test out all 9 possible combinations, and instead the three versions of direction and distance encoding were subsequently combined. This resulted in three overall visualizations: circles with size, 0° lines with color, and helplines with both. (see Figure 3).

We used a questionnaire to measure the variable perception in the periphery. We asked participants how many object representations they perceived per run if they focused on a cross in the center of the slides. This was our perceptibility dependent variable. Performance (for direction and distance) was measured through paper-based responses, where participants had to indicate on sheets of paper with diagrams. First, we asked for vertical direction towards the object, second for horizontal direction towards the object and third for distance to the object. Vertical and horizontal directions were measured by binning responses into 30° range categories (see Figure 5a and 5b). For example a participant could say the vertical direction is between 30 to 60° and the horizontal direction is between 150 to 180°. Distance was measured in four classes: very near, near, far, very far (see Figure 5c). Before the test session, each participant was given a tutorial to get familiarized with these distance classes.

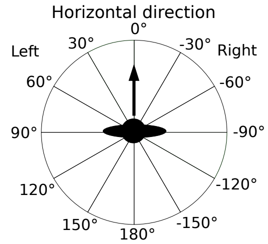
For this study, we asked: **RQ1: Which EyeSee360 visualization concept performs best with respect to accuracy for direction and distance towards the out-of-view object, and can be perceived well in the periphery?** Our dependent variables were: perceptibility (the amount), horizontal direction, vertical direction, and perceived distance.

Given our concept definition and study setup, we posit the following hypotheses: H_1**: Direction encoding with helplines results in better user performance than without helplines.** H_2**: Distance encoding results in better user performance with combination than with color or with size.**

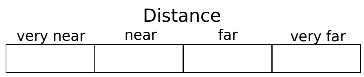
To avoid learning effects or fatigue, we counterbalanced our independent variables. Using a Latin square design, we arrive at 3 rows for the study. For each visualization combination, 10 slides where prepared. This corresponds to a total number of 30 slides. Since our rapid prototyping approach requires a change of the



(a) Vertical direction form



(b) Horizontal direction form



(c) Distance form

Figure 5: Paper-based forms to collect user input.

slides, we limited each condition to only 4 out of the 10 possible slides (which are variations of that condition). The choice of these 4 slides was chosen randomly.

4.5 Procedure

At the start of the study, participants received an introduction to out-of-view objects and were given a demo where they could look at the different visualization techniques. All possible variants of the visualizations were explained in detail using the three slides in Figure 3. As already mentioned, the order of the visualizations was counterbalanced. The three different visualizations were then applied successively to the AR prototype during the main part of the study.

First, participants were asked how many out-of-view object representations can be seen if s/he concentrates on the cross at the center of the slide. The maximum number of representations that can be seen on each slide is five. Next, the experimenter, together with the participant, walked through each numbered representation starting at one to five, asking for the three necessary values: vertical direction, horizontal direction and distance. As mentioned earlier, the used forms can be seen in Figure 5. After all visualizations were completed, participants filled out an SUS questionnaire, followed by a questionnaire with four questions concerning the implementation of the visualizations. At the end of the study, participants filled out a personal information form (age, gender, experience with off-screen objects rated on a 5-point Likert-scale, where 1 is strongly disagree). Each session lasted between approximately 60-75 minutes.

4.6 Participants

We recruited 19 participants[4] (8 females), aged between 20 and 61 years (M=28.3, SD=11). None suffered from color vision impairments, and all had normal or corrected vision. 11 had no experience with off-screen objects, and eight were somewhat familiar with such visualizations (Md=1, IQR=1-2).

4.7 Results

Perceptibility We consider the effects of three different conditions (size, color, combination) on object perceptibility. On each slide, five out-of-view objects were shown to the participant. To measure the perception performance, the user was asked to focus on a cross in the center of the slide and to state the number of representations for out-of-view objects that are visible. This helped us to see if a representation was too small or a color could not be perceived. The mean perceived number of objects (Max=5) for 'Ellipse with size'=4.91, '0° lines with color'=4.63 and 'Helplines with combination'=4.62. Normality here was not assumed because the Shapiro-Wilk test was significant ($p < 0.001$). We therefore ran a Friedman test, which revealed a significant effect of different encodings on perception error ($\chi^2(2)$=21.24, $p < 0.001$, N=19). A post-hoc test using Wilcoxon Signed-rank with Bonferroni correction showed significant differences between some conditions, which are shown in Table 2.

Comparison	P-value	ϕ-value
Ellipse with size + 0° lines with color	< 0.001	0.33
Ellipse with size + Helplines with combination	< 0.001	0.36
0° lines with color + Helplines with combination	0.85	0.03

Table 2: Pairwise comparison of perceptibility conditions.

Direction Accuracy We consider the effects of three different variants of our EyeSee360 technique ('Ellipse with size', '0° lines with color', 'Helplines with combination') on vertical and horizontal direction accuracy. It is important to know that we divided the different degree values in classes or buckets, where each bucket represents 30° of angle. These buckets were used to simplify participant input entry. For example if the out-of-view object's position is within 0° and 30° horizontally and the user thinks it is located between 60° and 90°, the error will be 2 because the guessed class is two points away from the correct one.

For vertical we had 6 (90° up, 90° down) and for horizontal we had 12 buckets (360° surround). The mean errors for vertical direction are 'Ellipse with size'=0.19, '0° lines with color'=0.15 and 'Helplines with combination'=0.12 (max. error possible 5). The mean errors for horizontal direction are 'Ellipse with size'=0.63, '0° lines with color'=0.51 and 'Helplines with combination'=0.28 (max. error possible 6). For vertical direction error, a Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$), and thereafter we ran a Friedman test that revealed no significant effect of different encodings on vertical distance error ($\chi^2(2)$=2.98, p = 0.225, N=19).

For horizontal direction error, a Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$), and thereafter we ran a Friedman test that revealed significant effects on all encodings for horizontal error ($\chi^2(2)$=48.27, $p < 0.001$, N=19). A post-hoc test using Wilcoxon Signed-rank with Bonferroni correction showed significant differences for some conditions (see Table 3).

Here, the pairwise comparisons show that the lower mean error for horizontal direction for 'Helplines with combination' is significant compared to 'Ellipse with size' and '0° lines with color'. In

[4]To obtain sufficient power for our data, we need 18 participants in our study. We calculated this value with G*Power under two-way ANOVA (α = 0. 05 and $1-\beta$ = 0.8) based on the three different variants of visualization techniques. For the perception in the periphery we get 216 data points because we get this based on the slides and not on every object (f = 0.22). For direction and distance towards out-of-view object we get 1080 data points and we are able to show small effect sizes of (f = 0.1).

Comparison	P-value	ϕ-value
Ellipse with size + 0° lines with color	0.076	0.06
Ellipse with size + Helplines with combination	< 0.001	0.24
0° lines with color + Helplines with combination	< 0.001	0.19

Table 3: Pairwise comparison of help variations for horizontal direction.

Figure 6 the horizontal and vertical direction error is plotted for all three variants combined. It is visible that most of the error of the paritcipants is limited to error class 1, which means guessing the direct neighbour of the correct bucket.

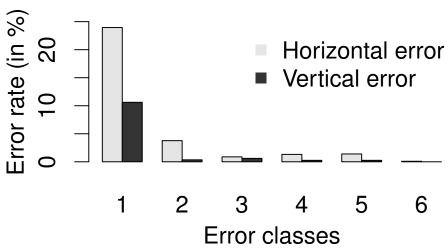


Figure 6: Barplot on error rate of horizontal and vertical direction error.

Distance Accuracy We investigated the effects of three different Attribute conditions ('Ellipse with size', '0° lines with color', 'Helplines with combination') on distance accuracy. Here again, we created a class- or bucket-based encoding for distance: very near, near, far, very far. The mean errors for distance are 'Ellipse with size'=0.15, '0° lines with color'=0.07, 'Helplines with combination'=0.09 (max. error possible is 3). A Shapiro-Wilk-Test showed that they are not normally distributed ($p < 0.001$), and a Friedman test revealed significant effects on the encodings for distance ($\chi^2(2)$=20.34, $p < 0.001$, N=19). A post-hoc test using Wilcoxon Signed-rank with Bonferroni correction showed significant differences for some conditions (see Table 4).

Comparison	p-value	ϕ-value
Ellipse with size + 0° lines with color	< 0.001	0.14
Ellipse with size + Helplines with combination	< 0.001	0.12
0° lines with color + Helplines with combination	0.618	0.02

Table 4: Pairwise comparisons of help variations for distance.

System Usability Scale For this concept validation study, our rapid prototype of EyeSee360 scored 71 on the SUS, which is just on the threshold for acceptable usability [3].

User Feedback To gain more specific feedback, we gave participants four 5-point Likert-scale questions (1 - strongly disagree, 5 - strongly agree). Participants could not strongly distinguish the colors used for distance encoding (Md=4, IQR=4-5), could strongly distinguish the sizes used for distance encoding (Md=4, IQR=3-4), did find the inner elipse of the visualization confusing (Md=2, IQR=1-2), and found the helplines in the visualization helpful for understanding the direction encoding (Md=5, IQR=4-5). Participants reported that the color yellow is not very perceptible, especially when shown on the periphery. This is in line with our findings of the perceptibility of representation in our early prototype that use the yellow color. Another point reported by users was that two very close out-of-view objects can not be encoded by our visualization because one can occlude the other.

4.8 Discussion

In validating our lo-fi EyeSee360 prototype, a number of issues arose. First, helplines were deemed to be the most useful by participants, and also had the lowest error on direction estimation, which fed into later design iterations. Therefore, we can accept our hypothesis H_1. Second, color and the combination of color with size overall were the best attribute for distance encoding, as shown from our quantitative results. But since there was no significant difference between color and color with size we can not accept our hypothesis H_2. Finally, the SUS scores indicated acceptable perceived usability for this lo-fi prototype, which provided further support for implementing our final EyeSee360 prototype.

5 PART II: EYESEE360 EVALUATION STUDY

Given the promising results of the Concept Validation study, we implemented the best performing version of our novel EyeSee360 visualization technique. This was done for two reasons: First, we are now able to evaluate this technique in an actual implemented prototype, which increases ecological validity. Second, we can then compare our new visualization technique with the adapted versions of three well-known 2D off-screen visualization techniques (Arrow, Halo and Wedge) for encoding the direction towards out-of-view objects [8]. Additionally, we evaluated the distance encoding of EyeSee360. We further investigate the subjective workload incurred from all tested visualization techniques and asses their usability with the SUS score. To evaluate the scalability of EyeSee360 we tested with different Number of Objects (three vs. five vs. eight).

5.1 Implementation

All visualizations here were implemented for Google Cardboard, where the video see-through variant was used. Vuforia was additionally used for the implementation to keep the out-of-view objects at fixed position in the environment. We used the Vuforia environment tracking based on the Gyroscope sensor of the smartphone. Our Google cardboard used for the evaluation had a field-of-view of 45° horizontal and 30° vertical. Our development was done in Unity and our implementation of EyeSee360 supports various devices (Google Cardboard, Microsoft Hololens, Oculus Rift etc.)[5].

Our validation study showed that a visualization with 45° helplines is best suited to recognize the rotation direction of out-of-view objects (Variant 3: Helplines with both; cf., Figure 3c). The representation of distance through colors and color and size were shown to be the best performing variants. To avoid too many conditions we chose to use only color since it revealed the smallest mean error. However, we found that showing yellow in the periphery was not

[5] https://github.com/UweGruenefeld/EyeSee.

very perceptible. For this reason, a color gradient from blue to red was chosen based on the cold and warm metaphor used for example in heatmaps[6] [11]. Here, red stands for very close and blue for far away.

In addition, the EyeSee360 visualization has to now represent objects dynamically through their representations. This means that the image has to change depending on the user's viewing direction. Furthermore, the previous visualization must be adapted to the smartphone. Here, the field of view was represented by an inner elipse. Due to the video see-through variant however, the camera image is looped through the device and output on the screen. Since this output no longer has a round shape but a rectangular shape, the inner area of the visualization must be adapted accordingly to also become rectangular. In other words, the looped camera image is the new field of view and since this image is retangular the focus area needs to be rectangular too. A further adaptation is that this focus area has to move when the user looks up or down, that is, in the vertical direction as the users adjusts her/his field of view. In these cases, the inner visualization moves along.

5.2 Study Design

The second study was also designed as a lab study and took place in the same empty office room with the same light conditions. The study is split into two parts, where the first aim is to compare the direction encoding of EyeSee360 with Arrow, Halo and Wedge (Part 1), and subsequently to evaluate the distance encoding for EyeSee360 (Part 2). Further we measure the SUS scores and the subjective workload for the different visualization techniques. For this study, we had to fix three parameters: (1) first, all objects were randomly distributed out of view in 3D space with equal possibilities for every possible direction and distance (ecxept the 3D space within the user's view) (2) Second, distance was ranged between 0% and 100% with one meters distance at 100% (3) Third, the position of the out-of-view objects are world-fixed, however not ego-fixed (or user fixed). For this study, we ask: **RQ2: Which visualization (Arrow, Halo, Wedge, EyeSee360) for head-mounted AR results in the best user performance with respect to direction, usability and workload?**

In the first part of the study, we investigate if the dependent variable Direction Accuracy is influenced by the independent variables Visualization (Arrow vs. Halo vs. Wedge vs. EyeSee360), Environment (180° vs. 360°) and Number of Objects (three vs. five vs. eleven). This repeated-measures within-subjects factorial design results in 24 different conditions. Additionally, we investigated if the dependent variables Usability and Workload are influenced by the independent variable Visualization (Arrow vs. Halo vs. Wedge vs. EyeSee360). In the second part of the study, we investigate if the dependent variables Distance Accuracy and Search Time are influenced by the independent variables Visualization (EyeSee360), Environment (360) and Number of Objects (three vs. five vs. eleven). This repeated-measures within-subjects factorial design results in 3 different conditions.

Given our concept definition and study setup, we posit the following hypotheses:

[6] https://en.wikipedia.org/wiki/Heat_map

H_1**: EyeSee360 performs better than Arrow, Halo, and Wedge visualizations for estimating direction for all environments (180°, 360°).**

H_2**: The measured workload for EyeSee360 is higher than for the visualizations Arrow, Halo and Wedge.**

5.3 Procedure

At the beginning of the study, participants got an introduction to out-of-view objects and were given a demo where they could test the different visualization techniques.

Part 1: Direction Estimation In the first part, participants had to locate out-of-view objects under two different areas of environments (180° vs. 360°). These ranges are 180° ahead of the user or 360° around the user. The order of visualization, number of objects, and environment were presented according to a Latin square design, followed by randomization. The out-of-view objects were created randomly in the corresponding area of the environment (180° vs. 360°). As mentioned earlier, none of the objects occupied the user's viewport. Each combination of visualization, number of objects and environments was tested in 3 iterations. In each iteration, three representations were chosen randomly and successively highlighted green and then the participant had to guess the position of the according object to the representation without seeing the object itself.

To accomplish this, the participant had a green cursor in the center of the screen and a remote controller to confirm the current cursor position as the out-of-view object's position. The cursor could be moved by head movement. To avoid getting the exact position of an out-of-view object in a given technique through head movement, the visualization technique was only visible in a small area directly in front of the participant. Moving the green cursor out of a black circle disabled the visualization technique and the participant had to guess the out-of-view object's position by the affordances the technique offered (see Figure 7). The direction error is measured as angle between the position specified by the participant and the actual position of the object. We had to change the input method from verbal reporting in the concept evaluation study to digital input to enable participants to specify the position of out-of-view with higher accuracy. Further, in pilot trials participants stated that writing on a paper while looking through a video-see-through device feels uncomfortable.

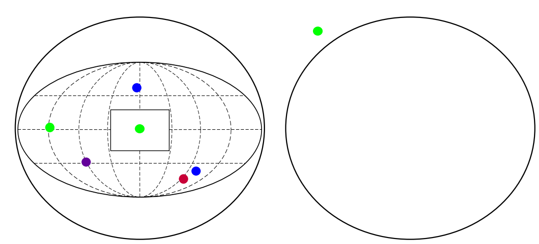


Figure 7: Left: green cursor is on the inside of the black circle and EyeSee360 is visible. Right: green cursor is on the outside of the black circle and EyeSee360 is not visible. *Best seen in color.*

After each visualization, users had to fill out a SUS[3] and a NASA-TLX[12] questionnaire about the used visualization technique.

Part 2: Distance Estimation and Search task In the second part, we evaluated the distance encoding for EyeSee360. Additionally, participants were asked to find the out-of-view object. The order of different number of objects were randomly presented. The out-of-view objects were created randomly in 360° around the user. Again, none of the objects occupied the user's viewport. Each number of objects was tested in 2 iterations. In each iteration, three representations were chosen randomly and successively the participant had to do the following two tasks with each chosen representation: (1) The chosen representation is highlighted, and the user had to estimate the distance towards the represented out-of-view object by moving a three-dimensional object on an axis extending from the user into the viewing space. The object was moved with a remote controller (2) Once the distance is entered, the highlighting is canceled, and the user's task is now to locate the represented object. This is accomplished by using the green cursor in the center of the screen to select one of the now visible out-of-view objects. The selection was confirmed with a remote controller.

At the end of the study, participants filled out a personal information form (age, gender and rated their experience with out-of-view objects and head-mounted devices on a 5-point Likert-scale, where 1 is strongly disagree and 5 is strongly agree). Each session lasted between approximately 45-60 minutes.

5.4 Participants

We recruited 16 participants[7] (8 females), aged between 20 and 63 years (M=30.6, SD=12.7) Most participants did not have much experience with visualizations of off-screen objects in 3D space (e.g., from video games) (Md=1, IQR=1-2), nor with head-mounted devices (e.g., AR or VR) (Md=1, IQR=1-1).

5.5 Results

Direction Accuracy We consider the effects of the three factors (Visualization, Number of Objects, Environment) on mean direction error. The mean errors for the visualization techniques are: Arrow=31.23°, Halo=29.74°, Wedge=28.52° and EyeSee360=21.25°. The direction errors are compared in Figure 8.

A Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$), and thereafter we ran a Friedman test that revealed a significant effect of visualization technique on direction error ($\chi^2(3)$=27.55, $p < 0.001$, N=16). A post-hoc test using Wilcoxon Signed-rank with Bonferroni correction showed significant differences between the four groups (see Table 5). EyeSee360 has a significantly lower direction error than Arrow and Halo. Furthermore, in Figure 8 it is visible that the standard deviation for EyeSee360 is the smallest.

Environment: The mean direction errors for environment are 24.06% for 180° and 31.31% for 360°. As we compare two matched groups within subjects, we directly performed a Wilcoxon Signed-rank test. Here we found a significant effect of environment (W = 241170, Z = -8.048, $p < 0.001$, $\phi = 0.17$). We further looked into effects of environment on each technqiue. We did a Wilcoxon Signed-rank test for Arrow (W = 12644, Z = -5.7713, $p < 0.001$, $\phi = 0.17$), Halo (W = 17866, Z = -2.0798, $p < 0.05$, $\phi = 0.12$), Wedge (W = 14258, Z = -4.6304, $p < 0.001$, $\phi = 0.14$) and EyeSee360 (W = 15216, Z = -3.9531, $p < 0.001$, $\phi = 0.12$). These were all shown to be significant.

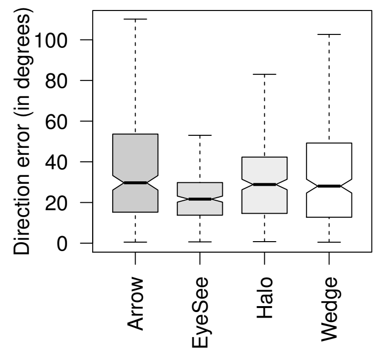


Figure 8: Boxplot of mean direction error for visualizations.

Comparison	P-value	ϕ-value
Halo, Arrow	0.723	0.01
Wedge, Arrow	0.053	0.06
EyeSee360, Arrow	< 0.001	0.20
Wedge, Halo	0.280	0.03
EyeSee360, Halo	< 0.001	0.20
EyeSee360, Wedge	< 0.001	0.15

Table 5: Pairwise comparisons of visualization techniques.

Number of Objects: Further, we investigated if the number of objects has a significant effect on the direction error. We compare three matched groups within subjects with a non-parametric Friedman test, which revealed no significant effect of number of objects on direction error ($\chi^2(2)$=3.2604, p = 0.20, N=16).

Subjective Workload According to hypothesis H2, we expected that EyeSee360 would have a higher subjective workload than Arrow, Halo and Wedge. Here, we compared the four matched groups (Workload scores for Arrow: 47.6, Halo: 42.5, Wedge: 44, EyeSee360: 46.6) within subjects with a non-parametric Friedman test. The Friedman test revealed no significant effect of visualization on NASA-TLX ($\chi^2(3)$=0.43, p = 0.93, N=16). This means we can not accept or reject the null hypothesis of *H2* that there is no difference in overall workload across techniques.

Distance Accuracy We compare three matched groups (3, 5, 11 objects) within subjects with a non-parametric Friedman test, which revealed no significant effect of number of objects on distance error in EyeSee360 ($\chi^2(2)$=1.8788, p = 0.39, N=16)). The distance was measured in a range between 0% to 100%.

Search Time Additionally, we investigated the search time for EyeSee360. In 297 runs, participants found 251 out-of-view objects

[7] The number of participants was calculated with G*Power for ANOVA. For direction we have 24 combinations. We assumed a alpha of 0.05 and a power of 0.8 (α = 0.05, 1 - β = 0.80). We need at least 14 participants to measure mean effect sizes (f = 0.25). For distance we again assumed a alpha of 0.05 and a power of 0.8 (α = 0.05, 1 - β = 0.80). With 14 participants we can also measure mean effect sizes of (f = 0.25).

correctly (success rate of 84.5%). Furthermore, the time needed to locate the out-of-view objects is important, where the mean time was 10.4 seconds. The mean time of all correct located objects is 9.8 seconds. Both values are quite high due to head rotation, selecting the out-of-view object and confirm the input. As a next step we divided the necessary rotation to find the object into classes (containing 30° each). In Figure 9, it is visible that the time needed to find an out-of-view object is increasing for higher angles except for 91° to 120°.

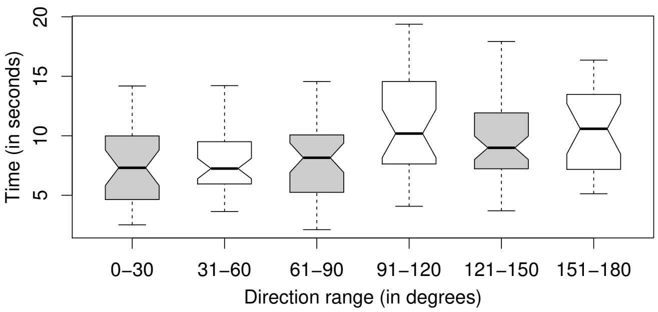


Figure 9: Median time for horizontal direction classes.

System Usability Scale To evaluate the perceived usability of EyeSee360, the SUS questionnaire was used. The result for the final prototype of EyeSee360 (68) is above Halo (66) and under Wedge (70). Compared to the SUS score for the rapid prototype version (71), it decreases. This is likely due to the fact that we used video see-through AR for the final prototype. Latencies with looping the camera to the screen can cause simulator sickness and worse usability.

Learning effects The number of trials and the time needed by participants is insufficient to investigate learning effects within our data. Therefore, to gain more specific feedback, we gave participants two 5-point Likert-scale questions (1 - strongly disagree, 5 - strongly agree) to investigate this. Participants felt they were able to understand the position of out-of-view objects visualized with EyeSee360 faster over time (Md=4, IQR=3-4), as well as more precisely (Md=4, IQR=4-4).

5.6 Discussion

In our final evaluation study of EyeSee360, we have a number of noteworthy findings. First, it appears that EyeSee360 had the lowest error for direction estimation, in contrast to the adapted 2D techniques. A pairwise comparison revealed that EyeSee performs significantly better than Arrow and Halo, but not siginificantly better than Wedge. Therefore, we can not accept or decline hypothesis H_1. Second, the 180° environment setting resulted in better direction accuracy than the 360° condition, which fits our intuitions that estimating direction for a larger spatial area is more difficult than a narrower range. The number of objects did not have any significant difference.

For workload, we also do not find any differences among the tested techniques and therefore, can not accept or reject hypothesis H_2. However, we do see an increasing trend in search task time as direction angle increases. This can have detrimental effects on workload in a real-life scenario. With respect to distance estimation, we found that there were no differences across the different number of objects, however the distance error was in the 5-10% range (M=9.34, SD=10.26) for all levels. Finally, it seems participants feel they do get better over time, despite that this was not explicitly tested.

6 GENERAL DISCUSSION

Advantages of AR Basically, EyeSee360 was inspired by 2D off-screen visualization techniques such as EdgeRadar [10] and therefore, is somehwat similiar to these techniques and familiar. For that reason EyeSee360 is using a plane projection in the view frustum of the user. Combined with a head-mounted device, our techniques offers a constant flow of information regarding out-of-view objects in the periphery.

From video to optical see-through AR or VR Another issue that arises in our work is whether our findings can transfer to other head-mounted AR devices (e.g., optical see-through). To do so however, measurement of the user's facial field is necessary when using optical-see-through devices. By contrast, for devices with video-see-through technology (as we have done), the facial field is determined by the camera lens used and therefore easier to determine. While for this study we were concerned with video see-through AR technology, we can see the potential of extending this work towards other AR and especially to Virtual Reality (VR) environments (since everything is rendered digitally in such an immersive environment).

Usefulness of Lo-fi prototyping tool From our first study, we developed a lo-fidelity protoyping tool to quickly test design ideas for out-of-view objects for head-mounted AR. We argue for the usefulness of this approach, as it saves both development time and allows designers to test and iterate quickly on user feedback. This is especially applicable to a visual domain such as out-of-view objects, were the exact parameters for size, color and form can vary even between users (e.g., color blind individuals).

Ecological Validity It is important to reflect on whether our developed techniques can be used in a real-world scenario. For example, ship docking [20] or gaming environments for social awareness [23]. While this was out of our current scope, the ultimate test of how well our EyeSee360 system can support users would require longitudinal in-the-wild testing, wherein we can gain greater insight into the interaction between learning effects, errors, and the specific context a head-mount AR device is used in (e.g., while mobile).

Study Limitations We did not measure the task completion times for the direction accuracy in the second study comparing Arrow, Halo, Wedge and EyeSee360 as this was not deemed relevant for answering our initial research question, and rather to focus on whether users can identify the object, and not do so as quick as possible. This can be addressed by future work when timing to locate out-of-view objects is crucial. Further, one could argue that the method of having to rotate the head of the user to identify the direction negativly affects the estimation of particpants on direction. But alternative modes such as having the user click on a 3D sphere or point in a direction have similar limitations.

7 CONCLUSION AND FUTURE WORK

In this paper we studied in depth the problem of visualizing out-of-view objects in head-mounted AR. Therefore, through a concept validation study and a comparative evaluation study, we proposed our EyeSee360 technique. This was shown to outperform the 2D adapted techniques from previous work. Further work should focus on reducing direction and distance error. Additionally, further comparative studies are needed for example to evalute EyeSee360 against AroundPlot[14]. Together, our methods and findings provide the groundwork by which future research on out-of-view object visualization in head-mounted AR can build on, particularly optical see-through (e.g., HoloLens) or Virtual Reality devices.

ACKNOWLEDGMENTS

We thank the Ministry of Science and Culture of Lower Saxony for supporting us with the graduate school *Safe Automation of Maritime Systems (SAMS)*.

REFERENCES

[1] Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: A Technique for Visualizing Off-screen Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 481–488. DOI: https://doi.org/10.1145/642611.642695

[2] Frank Biocca, Arthur Tang, Charles Owen, and Fan Xiao. 2006. Attention Funnel: Omnidirectional 3D Cursor for Mobile Augmented Reality Platforms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1115–1122. DOI: https://doi.org/10.1145/1124772.1124939

[3] John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

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

[5] Stefano Burigat, Luca Chittaro, and Silvia Gabrielli. 2006. 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)*. ACM, New York, NY, USA, 239–246. DOI: https://doi.org/10.1145/1152215.1152266

[6] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. 2009. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.* 41, 1, Article 2 (Jan. 2009), 31 pages. DOI: https://doi.org/10.1145/1456650.1456652

[7] Marco de Sá and Elizabeth Churchill. 2012. Mobile Augmented Reality: Exploring Design and Prototyping Techniques. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 221–230. DOI: https://doi.org/10.1145/2371574.2371608

[8] Uwe Gruenefeld, Abdallah El Ali, Wilko Heuten, and Susanne Boll. 2017. Visualizing Out-of-view Objects in Head-mounted Augmented Reality. In *Proceedings of the 19th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA. DOI: https://doi.org/10.1145/3098279.3122124

[9] Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: Clutter-free Visualization of Off-screen Locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 787–796. DOI: https://doi.org/10.1145/1357054.1357179

[10] Sean G. Gustafson and Pourang P. Irani. 2007. Comparing Visualizations for Tracking Off-screen Moving Targets. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07)*. ACM, New York, NY, USA, 2399–2404. DOI: https://doi.org/10.1145/1240866.1241014

[11] C.L. Hardin. 2000. Red and yellow, green and blue, warm and cool: explaining colour appearance. *Journal of Consciousness Studies* 7, 8-9 (2000), 113–122. http://www.ingentaconnect.com/content/imp/jcs/2000/00000007/F0020008/1046

[12] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology* 52 (1988), 139 – 183. DOI: https://doi.org/10.1016/S0166-4115(08)62386-9

[13] Niels Henze, Benjamin Poppinga, and Susanne Boll. 2010. Experiments in the Wild: Public Evaluation of Off-screen Visualizations in the Android Market. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM, New York, NY, USA, 675–678. DOI: https://doi.org/10.1145/1868914.1869002

[14] Hyungeun Jo, Sungjae Hwang, Hyunwoo Park, and Jung-hee Ryu. 2011. Aroundplot: Focus+ context interface for off-screen objects in 3D environments. *Computers & Graphics* 35, 4 (2011), 841–853.

[15] James Kalat. 2015. *Biological psychology*. Nelson Education.

[16] Adam M. Larson and Lester C. Loschky. 2009. The contributions of central versus peripheral vision to scene gist recognition. *Journal of Vision* 9, 10 (2009), 6. DOI: https://doi.org/10.1167/9.10.6 arXiv:/data/journals/jov/932854/jov-9-10-6.pdf

[17] Felix Lauber, Claudius Böttcher, and Andreas Butz. 2014. PapAR: Paper Prototyping for Augmented Reality. In *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)*. ACM, New York, NY, USA, 1–6. DOI: https://doi.org/10.1145/2667239.2667271

[18] Ji-Yong Lee, Joung-Huem Kwon, Sang-Hun Nam, Joong-Jae Lee, and Bum-Jae You. 2016. Coexistent Space: Collaborative Interaction in Shared 3D Space. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 175–175. DOI: https://doi.org/10.1145/2983310.2989181

[19] Kris Luyten, Donald Degraen, Gustavo Rovelo Ruiz, Sven Coppers, and Davy Vanacken. 2016. Hidden in Plain Sight: An Exploration of a Visual Language for Near-Eye Out-of-Focus Displays in the Peripheral View. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 487–497. DOI: https://doi.org/10.1145/2858036.2858339

[20] Marie-Christin Ostendorp, Jan Charles Lenk, and Andreas LÃijdtke. 2015. Smart Glasses to Support Maritime Pilots in Harbor Maneuvers. *Procedia Manufacturing* 3 (2015), 2840 – 2847. DOI: https://doi.org/10.1016/j.promfg.2015.07.775

[21] Benjamin I. Outram, Yun Suen Pai, Kevin Fan, Kouta Minamizawa, and Kai Kunze. 2016. AnyOrbit: Fluid 6DOF Spatial Navigation of Virtual Environments Using Orbital Motion. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 199–199. DOI: https://doi.org/10.1145/2983310.2989195

[22] James F Parker Jr and Vita R West. 1973. Bioastronautics Data Book: NASA SP-3006. *NASA Special Publication* 3006 (1973).

[23] Martin Pielot, Oliver Krull, and Susanne Boll. 2010. Where is My Team: Supporting Situation Awareness with Tactile Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1705–1714. DOI: https://doi.org/10.1145/1753326.1753581

[24] David Rudi, Ioannis Giannopoulos, Peter Kiefer, Christian Peier, and Martin Raubal. 2016. Interacting with Maps on Optical Head-Mounted Displays. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 3–12. DOI: https://doi.org/10.1145/2983310.2985747

[25] Manojit Sarkar and Marc H Brown. 1994. Graphical fisheye views. *Commun. ACM* 37, 12 (1994), 73–83.

[26] Torben Schinke, Niels Henze, and Susanne Boll. 2010. 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)*. ACM, New York, NY, USA, 313–316. DOI: https://doi.org/10.1145/1851600.1851655

[27] Teresa Siu and Valeria Herskovic. 2013. SidebARs: Improving Awareness of Off-screen Elements in Mobile Augmented Reality. In *Proceedings of the 2013 Chilean Conference on Human - Computer Interaction (ChileCHI '13)*. ACM, New York, NY, USA, 36–41. DOI: https://doi.org/10.1145/2535597.2535608

[28] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 265–272. DOI: https://doi.org/10.1145/223904.223938

[29] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. 2011. Peripheral vision and pattern recognition: A review. *Journal of vision* 11, 5 (2011), 13–13.

[30] Mengu Sukan, Carmine Elvezio, Ohan Oda, Steven Feiner, and Barbara Tversky. 2014. 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)*. ACM, New York, NY, USA, 331–340. DOI: https://doi.org/10.1145/2642918.2647417

[31] Marcus Tonnis and Gudrun Klinker. 2006. Effective Control of a Car Driver's Attention for Visual and Acoustic Guidance Towards the Direction of Imminent Dangers. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '06)*. IEEE Computer Society, Washington, DC, USA, 13–22. DOI: https://doi.org/10.1109/ISMAR.2006.297789

[32] Matthias Trapp, Lars Schneider, Christine Lehmann, Norman Holz, and Jürgen Döllner. 2011. Strategies for visualising 3D points-of-interest on mobile devices. *Journal of Location Based Services* 5, 2 (2011), 79–99. DOI: https://doi.org/10.1080/17489725.2011.579579

[33] Polle T. Zellweger, Jock D. Mackinlay, Lance Good, Mark Stefik, and Patrick Baudisch. 2003. City Lights: Contextual Views in Minimal Space. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 838–839. DOI: https://doi.org/10.1145/765891.766022