Beyond Halo and Wedge: Visualizing Out-of-View Objects on Head-mounted Virtual and Augmented Reality Devices

Uwe Gruenefeld,[1] **Abdallah El Ali**,[2] **Susanne Boll**,[1] **Wilko Heuten**[3]
[1]University of Oldenburg, [2]Centrum Wiskunde & Informatica (CWI), [3]OFFIS - Institute for IT
Oldenburg, Germany | Amsterdam, Netherlands | Oldenburg, Germany
{uwe.gruenefeld, susanne.boll}@uol.de, abdallah.el.ali@cwi.nl, wilko.heuten@offis.de

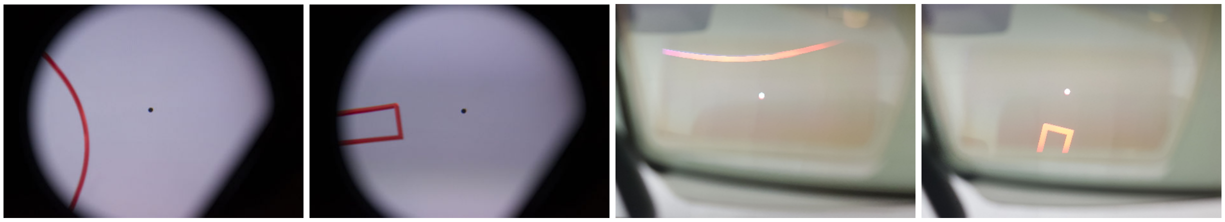

(a) *HaloVR*. (b) *WedgeVR*. (c) *HaloAR*. (d) *WedgeAR*.

Figure 1: **Implementation of *HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*. *Best seen in color.***

ABSTRACT

Head-mounted devices (HMDs) for Virtual and Augmented Reality (VR/AR) enable us to alter our visual perception of the world. However, current devices suffer from a limited field of view (FOV), which becomes problematic when users need to locate out of view objects (e.g., locating points-of-interest during sightseeing). To address this, we developed and evaluated in two studies *HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*, which are inspired by usable 2D off-screen object visualization techniques (Halo, Wedge). While our techniques resulted in overall high usability, we found the choice of AR or VR impacts mean search time (VR: 2.25s, AR: 3.92s) and mean direction estimation error (VR: 21.85°, AR: 32.91°). Moreover, while adding more out-of-view objects significantly affects search time across VR and AR, direction estimation performance remains unaffected. We provide implications and discuss the challenges of designing for VR and AR HMDs.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g., HCI): Miscellaneous

Author Keywords

Head-mounted; virtual reality; augmented reality; halo; wedge; out-of-view; off-screen; visualization technique

MobileHCI '18, September 3–6, 2018, Barcelona, Spain

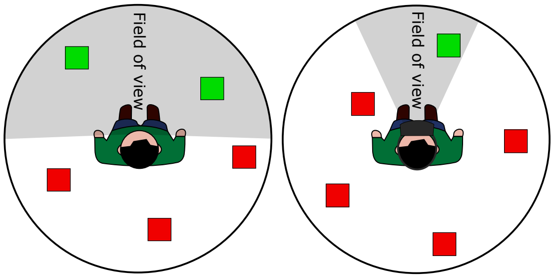
ISBN 978-1-4503-5898-9/18/09…$15.00
DOI: https://doi.org/10.1145/3229434.3229437

INTRODUCTION

Recent advances in Virtual Reality (VR) and Augmented Reality (AR) technology enable a variety of new applications (e.g., multi-player games in real environments [17]). What both technologies have in common is the ability to alter our perception of the world. Perception is altered by either adding virtual objects to our existing environment (AR) or by creating a complete virtual environment (VR). Combined with a tracked head-mounted device (HMD), VR and AR allow naturally changing the view dependent on head-movement, even while users are mobile. However, the problem of objects being hidden out of view still exists. Since the human field-of-view (FOV) is limited, spatially distributed objects outside of this range will not be perceived. This problem is amplified by a device's restricted FOV (Figure 2), which further decreases the human visual range (e.g., wearing a VR headset).

Out-of-view objects can be either virtual objects (e.g., opponents or enemies in multi-player games [19]) or real objects in the surrounding environment (e.g., points of interest during sightseeing [10]). To solve the problem of objects being hidden out of view, we use on-display VR and AR visualization techniques to augment the user's field of view. We add virtual objects pointing towards the position of out-of-view objects (which we call 'proxies'). Thereby, we can extend the range in which users can perceive the position of virtual or real objects to cover the 360° around the user. Furthermore, we aim to visualize out-of-view objects in the periphery of the user to avoid occlusion and maintain immersion. This approach allows users to constantly observe the position of out-of-view objects while their focus stays uncluttered.

In this paper, we developed four new visualization techniques (*HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*) that are inspired



(a) Natural field-of-view [12]. (b) Device field-of-view.

Figure 2: **Examples of out-of-view objects (red) and objects inside the field of view (green).** ***Best seen in color.***

by 2D mobile off-screen visualization techniques [1, 7], and applied them to VR and AR. In our recent work [5], we showed that these off-screen visualization techniques can be used for pointing towards out-of-view objects in AR. However, the use of a 2D overlay is not extendable for pointing to out-of-view objects distributed 360° around the user. Therefore, we developed our proxies as 3D objects pointing in the direction of out-of-view objects. We limited our proxies to visualize only the direction towards out-of-view objects, which is sufficient for bringing out-of-view objects into the user's field of view (e.g., showing a relevant point-of-interest during sightseeing) and therefore, well-suited as a first approach towards visualizing out-of-view objects in Virtual and Augmented Reality.

In this paper, we ask: **(RQ1) How fast can out-of-view objects be found and (RQ2) how accurately can their position be estimated using our developed techniques in Virtual and Augmented Reality?** To evaluate the performance of our visualization techniques (*HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*), we conducted two user studies. The first user study was done in VR using the Oculus Rift and the second study was done in AR using the Microsoft HoloLens.

Our research contributions include:

1. Development of four 3D out-of-view visualization techniques inspired by 2D off-screen visualization techniques for head-mounted VR (*HaloVR, WedgeVR*) and AR (*HaloAR, WedgeAR*).
2. An evaluation and thereby comparison of the developed techniques (*HaloVR/WedgeVR*, *HaloAR/WedgeAR*) for visualizing out-of-view objects in head-mounted VR and AR.

RELATED WORK

Off-Screen visualization on small screen devices

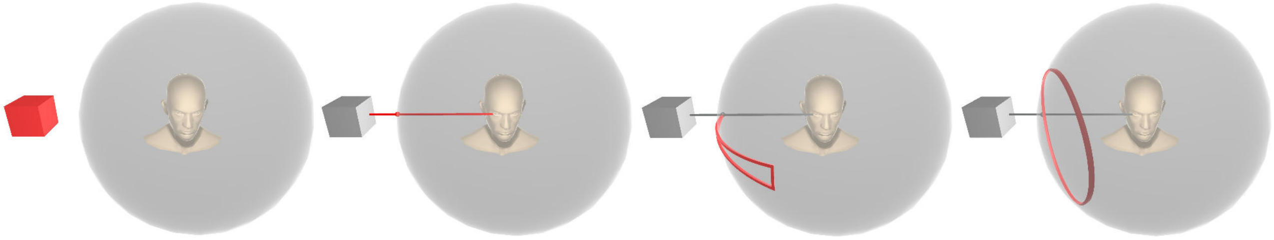
Since our work is inspired by existing off-screen visualization techniques, we cover a subset of research in that space. The three main approaches used to overcome small displays in 2D include: Overview+detail, Contextual views, and Focus+context [7, 4]. Contextual views and Focus+Context both overlay the screen borders with context information while Overview+Detail shows a miniature map of the surrounding area. A disadvantage of the miniature map is the cognitive load required to mentally integrate all views, while context information along the borders is more in line with the human frame of reference. Contextual views and Focus+Context differ in the kind of transition between focus and context. In the Focus+Context approach the transition is soft (e.g., fisheye-views that convey a distorted view [20]) and for Contextual views the transition is hard (e.g., arrows pointing into off-screen space [3]). Since Contextual views are distortion free due to the hard transitions, we utilized them in our work with head-mounted VR and AR.

One of the first Contextual views was presented by Zellweger et al. [23], who provided contextual information along the borders but users found it difficult to guess the actual 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 for Halo is cluttering, which is the accumulation of many Halos in corners. In Arrow the smaller shape of arrows is used to point towards off-screen objects. Several studies compared Halo with Arrow approaches [3, 11], where Arrows with fixed sizes performed worse than Halo while scaled arrows performed slightly better. Also the amount of visible objects have a high impact on the performance.

To avoid cluttering, researchers developed Wedge [7], which uses less space with isosceles triangles. However the smaller form can lead to an inaccurate understanding of the off-screen object's position. More recent work was looking into transferring existing off-screen visualization techniques to 3D space. For example Halo3D by Perea et al. [18]. They developed 3D Halos for mobile Augmented Reality applications but their approach does not consider head-mounted devices. Therefore, it visualizes off-screen objects in front of the user indistinguishable to off-screen objects behind the user. This is a problem for scenarios where off-screen objects are distributed 360°around the user (e.g., sightseeing or gaming) but is quite useful for their described scenario to visualize point of interests in industrial facilities that are mostly in front of the user.

Pointing towards out-of-view objects using peripheral displays

Nakuo and Kunze [16] presented an initial peripheral vision glasses prototype, that can display patterns in the periphery of the user's FOV. However, their prototype is limited in what can be shown in the left and right periphery and does not include different object positions, making it difficult to point towards out-of-view objects. Xiao et al. [22] presented SparseLight, introducing a matrix of LEDs placed in head-mounted VR and AR devices to create higher immersive experiences. They showed SparseLight's usefulness in conveying peripheral information and improving situational awareness, and reducing motion sickness. While we use proxies to indicate position of out-of-view objects, they use visual clones shown on multiple LEDs embedded on the HMD in an absolute mapping. This makes our approach more suitable for representing out-of-view objects irrespective of how distributed they are in a 180° environment. Luyten et al. [14] looked into visualization for near-eye out-of-focus displays, where they focused on specifying characteristics required to ensure good perceptibility. They found that simple shapes and a small set of colors



(a) Out-of-view object and user. (b) Intersection on sphere. (c) WedgeVR/AR. (d) HaloVR/AR.

Figure 3: **Projection of out-of-view object onto sphere with applied visualization techniques.** ***Best seen in color.***

are important for improving perception and comprehension of what is being shown on such displays.

Off-Screen visualization in console games

Aside from research, off-screen visualization is frequently used in computer games. In Rocket League (2015)[1] a 3D arrow is used for pointing towards the direction of a ball. However, more complex visualizations are also used when not only the 3D direction is required but also the 3D distance. In 3D games like Eve: Valkyrie (2016)[2], a radar-like visualization is used. However, since these commercial solutions do not offer any systematic user evaluations, we consider them more as a source of inspiration for our visualization techniques, rather than baseline measures.

Out-of-view object visualization

Lin et al. [13] investigated guiding gaze in 360° videos on smartphones. They presented two approaches for guiding attention in 360° videos: Auto Pilot (bringing target to viewers) and Visual Guidance (indicating direction of target). They showed that if increased head movement is necessary (e.g., following a sports video), users preferred Auto Pilot. Furthermore, users found it frustrating to shift to a target that is already gone. This highlights the need for accurate visualization of out-of-view objects. In our recent work [5], we adapted Arrow, Halo and Wedge to head-mounted Augmented Reality. Our results showed that all these techniques are applicable on head-mounted devices but our approach was limited to 90 degrees in front of the user. Therefore, we developed a new visualization technique called EyeSee360 [6], which is inspired by EdgeRadar [8]. However, EyeSee360 has a height workload and adds clutter to devices with a smaller field of view. Therefore, we chose Halo and Wedge as an inspiration for our techniques since they were the best performing techniques aside from EyeSee360.

GENERAL APPROACH

We explore the visualization of out-of-view objects in the 360° around the user. Our techniques utilize the periphery of the user to point towards out-of-view objects. However, our techniques are not constrained to the periphery so they can also be shown in the focus area. This makes sense if the head-mounted device has a small field-of-view which brings all displayed content to foveal vision (e.g., HoloLens). As mentioned earlier, our work draws on our recent work [5] that shows that out-of-view object visualization can be successfully adapted from 2D off-screen visualization techniques.

In this work, we wanted to explore how well our 3D visualization techniques can work across a) narrow FOV ranges and b) visual representations in AR and VR. Furthermore, we needed to evaluate our techniques for both technologies (VR, AR) because they have different influencing factors (e.g., different lighting conditions). For VR, we used the Oculus Rift because the display resolution and FOV are the current state of the art. The same applies to the HoloLens in AR, with the advantage of highly accurate placement of 3D objects in real environments. We address each of the VR and AR technologies in different subsections of our paper, where the first part describes our user study in VR and the second part does the same for AR.

DESIGNING OUT-OF-VIEW VISUALIZATION TECHNIQUES

A key aspect of our approach is the use of two off-screen visualization techniques (Halo [1], Wedge [7]) as an inspiration for our developed visualization techniques. Both techniques make use of well-known simple shapes that users can mentally complete even when only part of a shape is visible, a process known as *amodal completion* [15]. We choose Halo and Wedge because both techniques rely on the Contextual views approach [3]. Contextual views mostly represent only objects of interest and do so using simple shapes (proxies) [7]. These visualizations, or proxies, can be described as follows:

Halo. Halos surround off-screen objects with rings. The center of the ring is exactly at the off-screen object's position. The rings are just large enough to be on-screen.

Wedge. Wedges use isosceles triangles to represent the position of each off-screen object. The tip of the triangle is at the object's position. They make room for each other to avoid cluttering.

To visualize the 3D direction towards out-of-view objects with our techniques, we first need to obtain that direction from the object's position in 3D space. Therefore, we project the position of out-of-view objects onto an imaginary sphere around the user's head. This projection is done by drawing an imaginary line between the user's head and the position of the out-of-view object. The point where this line intersects with the imaginary sphere is the normalized vector representing the 3D direction towards the out-of-view object (see Figure 3a and

---

[1] `https://www.rocketleague.com`, last retrieved May 29, 2018

[2] `https://www.evevalkyrie.com`, last retrieved May 29, 2018

Figure 3b). Next, we obtain a normalized vector for each out-of-view object representing its 3D direction. Each normalized vector is located on the imaginary unit sphere around the user's head. As a second step, we project our techniques (*HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*) onto the inner surface of the imaginary sphere around the user's head (see Figure 3c and Figure 3d). We accomplish this by spanning our proxies between the intersection of the sphere (see Figure 3b) and the user's line of sight. The sphere we used has a virtual radius of one meter.

To fully utilize the existing space around the user, we show the proxies for out-of-view objects in all directions (including above the user's head) with a constant distance to the user's line of sight. While the distance to the user's line of sight can be adapted based on need (focus / small field-of-view, peripheral / large-field-of-view), the distance is always kept the same. For this approach, we derive encodings that are based on human perception characteristics [12, 21] (e.g., color perception and resolution) to ensure that every out-of-view object is adequately and equally perceived. To show the proxies to out-of-view objects in all directions has the further advantage in providing the user with a better understanding of the direction of an out-of-view object even if they do not understand the amount of head movement required to find it. In addition to the original Halo[1] and Wedge[7] implementations, we needed to adapt our techniques for VR and AR views:

All techniques. We added transparency to all techniques. Using this approach, we ensure no content is occluded by our proxies, and it further helps reduce clutter in cases of overlapping proxies.

HaloVR, HaloAR. For *HaloVR* and *HaloAR* we needed to change the way the technique points towards out-of-view objects. Originally, the object is in the center of the Halo but if we apply these techniques to the inner surface of a sphere, this would limit Halo to only show objects 90° away from the user's line of sight. To overcome this limitation, we span *HaloVR* and *HaloAR* in between the user's line of sight and the direction towards the out-of-view object (cf. Figure 3b). The techniques are shown in 1a and 1c.

WedgeVR, WedgeAR. For *WedgeVR* and *WedgeAR* we needed to remove the ability to make space for other proxies. There are two reasons for this: first, this approach avoids jumping wedges during head movements. In pilot testing users reported losing track of specific out-of-view objects when there was jumping involved. The jumping of wedges would happen because of the original approach of making space for other wedges because here, they are applied onto the inner surface of a sphere which is a double-curved geometry. Making space for other wedges combined with head-movement requires wedges to jump. In the original work this was not happening because the wedge were applied onto a 2D plane. Second, we can ensure that all proxies are always displayed with the same distance towards the user's line of sight and can be perceived equally well. The techniques are shown in Figure 1b and Figure 1d.

We implemented our techniques using the Unity[3] game engine. To ensure smooth surfaces and 3D shapes, we generated the meshes for all techniques during run-time. Furthermore, we allowed adapting the number of polygons used for each technique to ensure good rendering performance on different platforms. Additionally, we added a directional light source to simulate natural light effects like shadows and reflections. We used for each platform the same 'intrusion' level into the users' field of view (5° Hololens, 25° Oculus). More platform specific implementation details are described in the corresponding parts. All four techniques (*HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR*) are available as Unity packages under MIT License on GitHub[4].

PART I: EVALUATING HALOVR AND WEDGEVR

In this section, we provide an evaluation of the performance of our designed techniques *HaloVR* and *WedgeVR*.

Apparatus

For testing our techniques in VR, we used the Oculus Rift because the display resolution and FOV are the current state of the art. The FOV of the Oculus Rift is about 90 degrees. This allows for a large design space in how to apply our techniques. Therefore, we decided to place the proxies in the periphery of the user. This helps to avoid clutter and object occlusion. Further, it serves to maintain an immersive experience. Also, prior work [6, 5] has shown that the periphery of the user can be used for out-of-view visualization, and provides a better user experience. During pilot testing, we observed that an area of 50° in the focus of the user should remain unaffected by our visualization techniques. This allows the user to focus on their primary task while they could still observe moving out-of-view objects in the periphery sufficiently as a secondary task. Since we are interested in how well these techniques perform, we needed to test them at first in environments without other influencing variables. Therefore, we ran our study in an empty space (see Figure 4a). We used the Oculus Touch controller as an input device.

Study design

To evaluate the performance of our techniques (*HaloVR* and *WedgeVR*), we conducted a within-subjects controlled laboratory study in VR with the Oculus Rift. Our study had two independent variables, Visualization with two levels (*HaloVR* vs. *WedgeVR*), and Number of Objects with three levels (one vs. five. vs. eight). We varied the number of shown out-of-view objects to investigate the threshold for a maximum number of out-of-view objects, and because of prior work that showed Halo suffers from on-screen cluttering (cf., [3, 11]). Besides that the same number of out-of-view objects (one vs. five vs. eight) were investigated in previous work (cf., [5]).

We used quantitative methods to evaluate user performance, where our dependent variables were search time, object selection accuracy and direction error. Search time is the time users need to locate and select an out-of-view object in the scene while object selection accuracy specifies the number

---

[3] `https://www.unity3d.com`, last retrieved May 29, 2018

[4] `https://github.com/UweGruenefeld/OutOfView`

of objects user's selected correctly. The direction error here is the angular error, which is the angle between the user's assessment of the out-of-view object's position and the correct position in 3D space. Additionally, we gave participants the SUS [2] and RAW-TLX [9] questionnaires in order to gain insights into perceived usability and workload. To gain further insight into the user experience of all techniques, we additionally provided participants with Likert-scale items that asked about their performance during the study.

For this study, we asked: **Which visualization technique (*HaloVR*, *WedgeVR*) performs best with respect to search time, object selection accuracy, and direction error for different numbers of out-of-view objects in Virtual Reality?** We posit the following hypotheses:

H_1: **Higher number of out-of-view objects results in worse search time performance.**
H_2: **Based on the findings in [5] where Halo outperformed Wedge with respect to direction error, we hypothesize *HaloVR* to result in lower direction error.**
H_3: **Both techniques (*HaloVR* and *WedgeVR*) result in acceptable usability.**

Procedure

Our study was divided into two tasks. A search task and a direction estimation task. Both tasks were divided into two blocks (four blocks in total), where each block tested one technique (*HaloVR*, *WedgeVR*). We counter-balanced each block across all participants. Each technique was tested with one, five and eight out-of-view objects (see Section 'Study design'). Each number of out-of-view objects was tested five times. The amount of objects to test in a trial was selected randomly at run time. Objects were randomly distributed in 3D space as out-of-view objects. We stored the seeds of the position generation to test the same positions for each technique. However, with randomly shuffling the seeds from the previous technique we ensured that participants could not recognize the position of out-of-view objects tested with the previous technique. Overall, we tested four blocks with three different number of objects for five iterations, resulting in 60 trials per participant.

After each block of the search task, we asked participants to fill out SUS and NASA RAW-TLX questionnaires. At the end of all blocks, participants were asked to fill out our subjective and demographic questionnaire.

Task A: Search time

Each of the two blocks of this task started with a practice trial and an explanation of the visualization and the task the user had to perform. In each run of this task, a randomly chosen proxy (*HaloVR*, *WedgeVR*) was highlighted in red and the participant had to find the represented out-of-view object with the support of the proxy by selecting it with a cursor and press a button on the controller. The cursor was controlled by head-movement.

Task B: Direction estimation

Each of the two blocks of this task started with a practice trial and an explanation of the visualization and the task the user had to perform. In each run of this task, a randomly chosen proxy (*HaloVR*, *WedgeVR*) was highlighted in red and the participant had to estimate the represented out-of-view object by selecting the estimated direction with a cursor and pressing a button on the controller. The cursor was controlled by head-movement. The visualization technique was only visible when the user gazed into the starting direction. Any kind of head-movement towards the out-of-view object disabled the visualization technique (i.e., made it invisible). Each participant took approximately 40 minutes to finish the experiment.

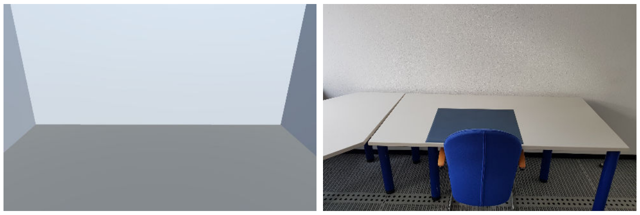

(a) Study setup in VR. (b) Study setup in AR.

Figure 4: **Study setup in Virtual and Augmented Reality. *Best seen in color.***

Participants

We recruited 16 participants (6 females), aged between 21 and 54 (M=30.06, SD=7.72). None of them suffered from color vision impairments. All had normal or corrected to normal vision. The participants did not receive any compensation.

Results

Search task

For the search task, we consider the effects of the two factors (Visualization, Number of Objects) on search time and object selection accuracy (where object selection accuracy means an object was not found during the trial). The mean search times for the visualization techniques are: *HaloVR*=2.26s and *WedgeVR*=2.24s. The total number of incorrectly selected objects are: *HaloVR* (232/240 = 96.7% accuracy) and *WedgeVR* (234/240 = 97.5% accuracy). The search times are compared in Figure 5.

A Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$). As we compare two matched groups within subjects, we directly performed a Wilcoxon Signed-rank test. Here we found no significant effect of visualization technique on search time (W = 14404, Z = -0.052, p = 0.959, $\phi < 0.01$). This indicates that *HaloVR* and *WedgeVR* do not sufficiently differ with respect to search time.

For a more detailed analysis we compared the different combination of visualization and number of objects (*HaloVR*(1)=1.96s, *HaloVR*(5)=2.31s, *HaloVR*(8)=2.52s, *WedgeVR*(1)=1.87s, *WedgeVR*(5)=2.18s, *WedgeVR*(8)=2.66s). Since our data is not normally distributed and we compare six matched groups within subjects, we ran a Friedman test, which revealed a significant effect of different combinations (visualization, number of objects) on search time ($\chi^2(5)=37.15$, $p < 0.001$, N=16). A post-hoc test using Wilcoxon Signed-rank

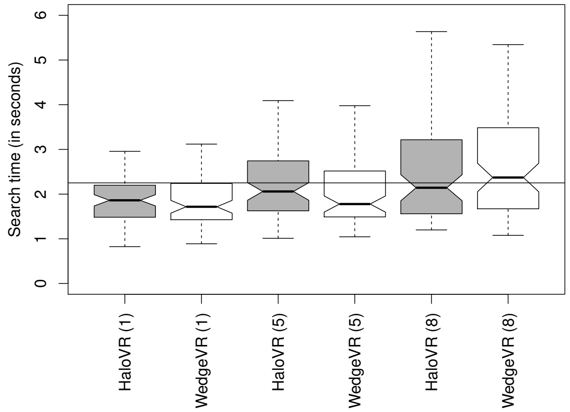


Figure 5: **Search times of visualization techniques (line indicates mean search time and number in brackets indicates number of objects tested).**

with Bonferroni correction showed significant differences between some conditions, which are shown in Table 1. We can conclude that the search time for one object is smaller than for multiple objects. Furthermore, for *WedgeVR* the search time is proportionally increasing, where *WedgeVR*(1) < *WedgeVR*(5) < *WedgeVR*(8).

Combination	**P-value**	**ϕ-value**
HaloVR(1) vs. *WedgeVR*(1)	1	undef.
HaloVR(5) vs. *WedgeVR*(5)	0.240	0.14
HaloVR(8) vs. *WedgeVR*(8)	0.200	0.10
HaloVR(1) vs. *HaloVR*(5)	0.003 **	0.23
HaloVR(1) vs. *HaloVR*(8)	< 0.001 ***	0.32
HaloVR(5) vs. *HaloVR*(8)	0.307	0.08
WedgeVR(1) vs. *WedgeVR*(5)	0.007 **	0.21
WedgeVR(1) vs. *WedgeVR*(8)	< 0.001 ***	0.40
WedgeVR(5) vs. *WedgeVR*(8)	< 0.001 ***	0.28

Table 1: **Pairwise comparison of technique combinations for search time performance (number in brackets indicates number of objects tested).**

Estimation task

We consider the effects of the two factors (Visualization, Number of Objects) on mean direction error. The mean errors for the visualization techniques are: *HaloVR*=23.02° and *WedgeVR*=20.67°. The direction errors are compared in Figure 6. A Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$). As we compare two matched groups within subjects, we performed a Wilcoxon Signed-rank test. Here we found no significant effect of visualization technique on direction error (W = 13165, Z = -1.203, p = 0.230, ϕ = 0.05). This indicates that *HaloVR* and *WedgeVR* do not significantly differ with respect to estimation accuracy.

Furthermore, we tested the Pearson's product moment correlation coefficient between direction error and angle towards out-of-view object. The direction error for *HaloVR* and *WedgeVR* can be seen in Figure 7. The correlation is *HaloVR*=0.663 (t(238)=13.677, $p < 0.001$) and for *WedgeVR*=0.710 (t(238)=15.543, $p < 0.001$). Our results indicate that there is a medium-strong correlation between direction error and angle towards out-of-view object for both techniques. This shows that a higher angle results in higher direction error.

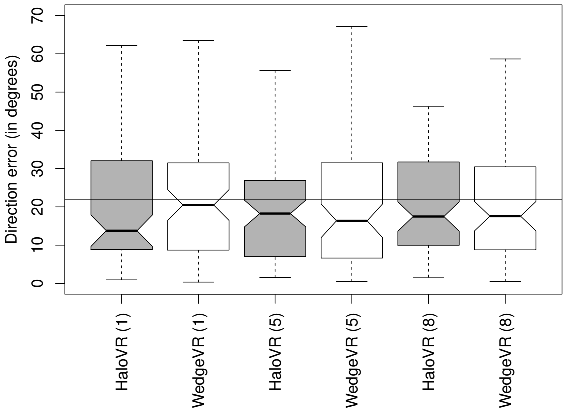


Figure 6: **Direction error of visualization techniques (line indicates mean direction error and number in brackets indicates number of objects tested).**

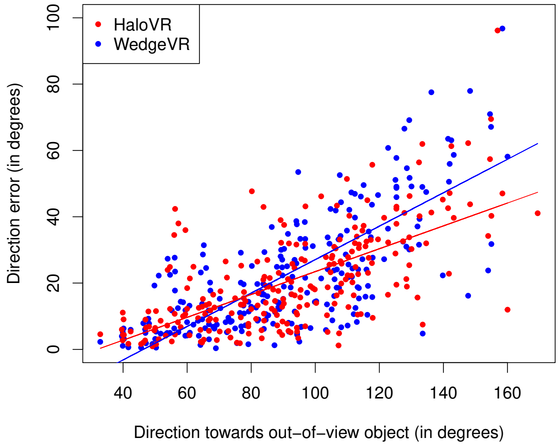


Figure 7: **Correlation between direction error and angle towards out-of-view object of *HaloVR* and *WedgeVR*. *Best seen in color.***

NASA RAW-TLX

For NASA Raw-TLX [9] scores, *HaloVR* scored 29.44 and *WedgeVR* scored 23.88. Both values indicate a low workload, while *WedgeVR* has a slightly lower workload than *HaloVR*. A t-test revealed no significant difference between the *HaloVR* (M=29.44, SD=13.23) and *WedgeVR* (M=23.88, SD=10.52) conditions; t(15)=2.154, p=0.048.

System Usability Scale

For SUS scores, *HaloVR* scored 87 and *WedgeVR* scored 92, both of which are over the threshold for acceptable usability [2]. This shows that both techniques are usable for VR.

Likert-scale questionnaire

At the end of the study, we asked participants to answer four questions with 5-point Likert items. Participants stated they were able to easily find the out-of-view objects with *HaloVR* (Md=4, IQR=1) as well as with *WedgeVR* (Md=5, IQR=1). Furthermore, they stated they were able to correctly estimate the position of out-of-view objects with *HaloVR* (Md=3, IQR=1.25) and with *WedgeVR* (Md=3.5, IQR=1). Overall, nine participants preferred *WedgeVR* while seven preferred *HaloVR*. Furthermore, three participants stated that it was harder to estimate the position than searching for the object. However all three of those participants stated that *WedgeVR* better supports position estimation than *HaloVR*.

Discussion

Number of objects. The number of out-of-view objects visualized simultaneously had a significant effect on the search time performance, while it had no effect on the direction error. This can be explained by participants who need more time when distraction is added by multiple proxies. Therefore, we suggest to use as few proxies simultaneously as possible. Since we found a significant effect of number of objects on search time performance, we can accept our hypothesis H_1.

Comparison of techniques. Both techniques performed well with respect to search time, object selection accuracy and direction error. In both tasks there were no significant differences between *HaloVR* and *WedgeVR*. Based on these findings, we can not accept our hypothesis H_2 because we did not find a significant effect for direction error. Both techniques had an acceptable usability (*HaloVR*=87 and *WedgeVR*=92) and a low workload (*HaloVR*=29.44 and *WedgeVR*=23.88), therefore we can accept H_3.

Clutter. Four out of 16 participants stated that they preferred *WedgeVR* for the study because it resulted in less clutter to their view. They further liked the aspect that *WedgeVR* always points directly in the direction of the out-of-view object.

PART II: EVALUATING HALOAR AND WEDGEAR

Here, we evaluate the performance of our designed techniques (*HaloAR*, *WedgeAR*) for foveal visualization of out-of-view objects in AR.

Apparatus

For AR, we used the Microsoft HoloLens since the placement of 3D objects in the real environment and the display technology used is currently the state of the art for HMD AR devices. Further, the limited field-of-view is quite challenging and is helpful to test how well our techniques generalize. The field-of-view of the Microsoft HoloLens is about 30 degrees. In comparison with VR, this reduces the available space for our visual proxies by a factor of three. On devices with this small field-of-view, showing our visualizations in the periphery is not possible. Therefore, we moved them to foveal vision, leaving only an area of 10 degrees unaffected. Further, we used a HoloLens Clicker instead of the Oculus Rift controller in the previous study.

Study design

To evaluate the performance of our designed techniques (*HaloAR* and *WedgeAR*) in AR, we conducted a second within-subjects controlled laboratory study with the Microsoft HoloLens. Our study in Augmented Reality was based on the same study design as our previous VR study.

For this study, we asked: **Which visualization technique (*HaloAR*, *WedgeAR*) performs best with respect to search time, object selection accuracy and direction error for different numbers of out-of-view objects in AR?** We posit the following hypotheses:

H_4: **Higher number of objects results in worse search time performance.**
H_5: **Based on findings in [5] where Halo outperformed Wedge with respect to direction error, we hypothesize that *HaloAR* outperforms *WedgeAR*.**
H_6: **We expect the smaller HoloLens FOV to negatively affect performance for search time, object selection accuracy and direction error even when the visualizations are presented in the focus instead of the periphery.**
H_7: **Both techniques (*HaloAR* and *WedgeAR*) result in acceptable usability.**

Procedure

Our study was divided into two tasks. A search task and a direction estimation task. Both tasks where similar to the tasks used for the VR study. We tested four blocks with three different number of objects for five iterations resulting in 60 trials per participant. Overall each participant took approximately 40 minutes to complete the study.

Participants

We recruited 16 participants (7 females), aged between 20 and 56 (M=30.63, SD=10.34). None of them suffered from color vision impairments. All had normal or corrected to normal vision. The participants did not receive any compensation.

Results

Search task

For the search task, we consider the effects of the two factors (Visualization, Number of Objects) on search time and object selection accuracy (where object selection accuracy means an object was not found during the trial). The mean search times for the visualization techniques are: *HaloAR*=3.84s and *WedgeAR*=4.00s. The total number of incorrectly selected objects are: *HaloAR* (235/240 = 98% accuracy) and *WedgeAR* (231/240 = 96.3% accuracy). The search times are compared in Figure 8.

A Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$). As we compare two matched groups within subjects, we performed a Wilcoxon Signed-rank test. Here we found no significant effect of visualization technique on search time ($W = 13161$, $Z = -1.2065$, $p = 0.228$, $\phi = 0.06$). This indicates that *HaloAR* and *WedgeAR* do not significantly differ with respect to search time.

For a more detailed analysis we compared the different combinations of visualization and number of objects (*HaloAR*(1)=3.05s, *HaloAR*(5)=3.57s, *HaloAR*(8)=4.91s,

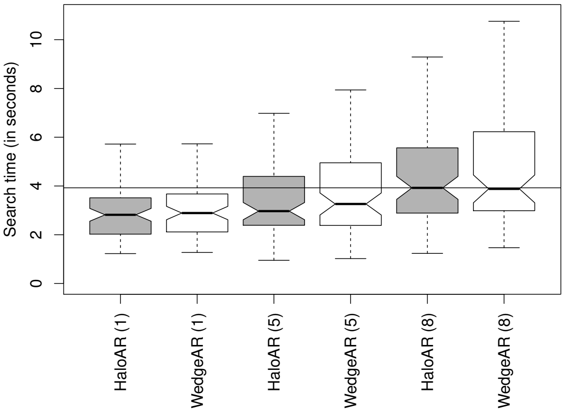


Figure 8: **Search times of visualization techniques (line indicates mean search time and number in brackets indicates number of objects tested).**

WedgeAR(1)=3.29s, *WedgeAR*(5)=3.97s, *WedgeAR*(8)=4.75s). Since our data is not normally distributed and we compare six matched groups within subjects, we ran a Friedman test, which revealed a significant effect of different combinations (visualization, number of objects) on search time (χ^2(5)=68.08, $p < 0.001$, N=16). A post-hoc test using Wilcoxon Signed-rank with Bonferroni correction showed significant differences between all conditions, which are shown in Table 1. We can conclude that the number of objects has a significant effect on search time.

Combination	**P-value**	**ϕ-value**
HaloAR(1) vs. *WedgeAR*(1)	0.869	0.13
HaloAR(5) vs. *WedgeAR*(5)	0.399	0.07
HaloAR(8) vs. *WedgeAR*(8)	1	undef.
HaloAR(1) vs. *HaloAR*(5)	0.025 *	0.18
HaloAR(1) vs. *HaloAR*(8)	< 0.001 ***	0.46
HaloAR(5) vs. *HaloAR*(8)	< 0.001 ***	0.31
WedgeAR(1) vs. *WedgeAR*(5)	< 0.001 ***	0.27
WedgeAR(1) vs. *WedgeAR*(8)	< 0.001 ***	0.39
WedgeAR(5) vs. *WedgeAR*(8)	= 0.001 **	0.25

Table 2: **Pairwise comparison of technique combinations on search time performance (number in brackets indicates number of objects tested).**

Direction Estimation task

We consider the effects of the two factors (Visualization, Number of Objects) on mean direction error. The mean errors for the visualization techniques are: *HaloAR*=29.79°and *WedgeAR*=36.03°. The direction errors are compared in Figure 9. A Shapiro-Wilk-Test showed that our data is not normally distributed ($p < 0.001$). As we compare two matched groups within subjects, we performed a Wilcoxon Signed-rank test. Here we found a significant effect of visualization technique on direction error (W = 11531, Z = -2.720, p = 0.006, ϕ = 0.12). This provides evidence that *HaloAR* results in significantly better performance than *WedgeAR* with respect to estimation accuracy. However, we found no significant differences between the six groups (*HaloAR*(1), *HaloAR*(5), *HaloAR*(8), *WedgeAR*(1), *WedgeAR*(5), *WedgeAR*(8)) (χ^2(5)=6.91, p = 0.228, N=16).

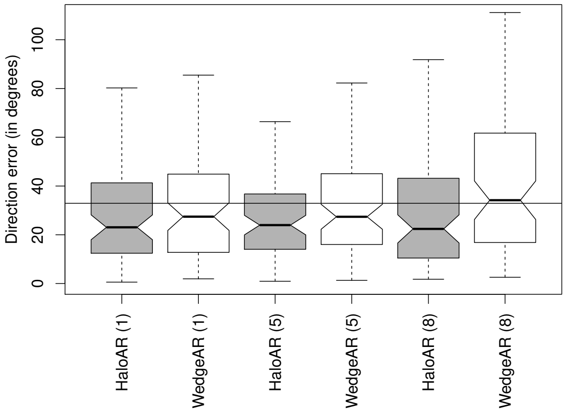


Figure 9: **Direction error of visualization techniques (line indicates mean direction error and number in brackets indicates number of objects tested).**

Furthermore, we tested the Pearson's product moment correlation coefficient between direction error and angle towards out-of-view object. The direction error of *HaloAR* and *WedgeAR* can be seen in Figure 10. The correlation is *HaloAR*=0.294 (t(238)=4.7504, $p < 0.001$) and for *WedgeAR*=0.372 (t(238)=6.1781, $p < 0.001$). Our results indicate that there is a weak correlation between direction error and angle towards out-of-view object for both techniques. This means a higher angle is less likely to result in a higher direction error than a lower angle.

NASA RAW-TLX

For NASA Raw TLX, *HaloAR* scored 21.75 and *WedgeAR* scored 24.81 [9]. Both values indicate a lower workload. A t-test revealed no significant difference between the *HaloAR* (M=21.75, SD=11.75) and *WedgeAR* (M=24.81, SD=14.57) conditions; t(15)=1.060, p=0.306.

System Usability Scale

HaloAR scored 85 and *WedgeAR* scored 82 on the SUS, which is above the threshold for acceptable usability [2]. According to the SUS scores, we find that both techniques are usable for AR HMDs.

Likert-scale questionnaire

At the end of the study, we asked the participants to answer four questions with 5-point Likert-scale items. Participants stated they were able to easily find the out-of-view objects with *HaloAR* (Md=4, IQR=2) as well as with *WedgeAR* (Md=4, IQR=1). Furthermore, they said they were able to easily estimate the position of out-of-view objects with *HaloAR* (Md=3, IQR=1.5) and *WedgeAR* (Md=3, IQR=2). Overall, six participants preferred *WedgeAR* while ten preferred *HaloAR*.

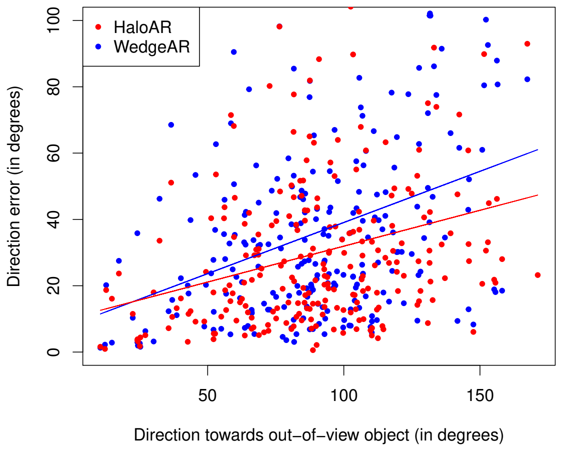


Figure 10: **Correlation between direction error and angle towards out-of-view object of *HaloAR* and *WedgeAR*. *Best seen in color.***

Discussion

Number of objects. The number of out-of-view object visualized simultaneously had a significant effect on the search time performance while it had no effect on the direction error. This can be explained by participants who need more time when distraction is added by multiple proxies. Therefore, we suggest to use as few proxies simultaneously as possible. Given our results, we accept hypothesis H_4.

Comparison of techniques. Both techniques performed well with respect to search time, object selection accuracy and direction error. We found a significant effect of visualization technique on direction error, but no significant effect of visualization technique on search time or object selection accuracy. To interpret this, the foveal presentation of *HaloAR* may have been easier to understand than *WedgeAR* and therefore possibly lead to lower direction estimation errors. Here, we reject our hypothesis H_5.

Furthermore, both techniques had an acceptable usability (*HaloAR*=85 and *WedgeAR*=82) and a low workload (*HaloAR*=21.75 and *WedgeVR*=24.81), therefore we can accept H_7.

Small field of view. The small field of view of the Microsoft HoloLens allows us to only use a small amount of out-of-view visualizations at the same time. This is due to frequent overlapping of multiple proxies that adds clutter to the screen even with our added transparency. This is supported by the significant increase in search time for *HaloAR*(5) to *HaloAR*(8) by 1.38 seconds and from *WedgeAR*(5) to *WedgeAR*(8) by 0.78 seconds. However, this affects *HaloAR* more strongly than *WedgeAR*. Here, we accept our hypothesis H_6.

IMPLICATIONS

Advantages of head-mounted devices. Since our techniques are inspired by off-screen visualization techniques [1, 7], they can be perceived as similar and therefore familiar to users. In this regard, our techniques also overlay the viewing frustum of the user. Combined with an HMD, our techniques offer a constant flow of information regarding out-of-view objects for presentation in both the periphery and on-demand guidance for foveal presentation under more limited FOVs such as AR.

Reducing the number of proxies. How many out-of-view objects should be visualized is dependent on the task. For estimating the position of out-of-view objects, the number of objects has no influence on user performance for up to at least eight objects. However, search time is negatively affected by a higher number of proxies across both VR and AR. Therefore, we recommend reducing the number of proxies that are visible at the same time during a search task. This can be achieved by successively guiding the user from one out-of-view object to the next one in a sequential manner, or through subsetting only relevant (e.g., determined by use-case) out-of-view objects at a given time.

Peripheral vs. foveal visualization. To adapt our techniques across different FOVs, we followed two different approaches to present our proxies to the user. For small FOV devices where no peripheral vision is available, we used a foveal presentation while for larger FOV devices, we presented the proxies peripherally. Based on our experiments, we recommend varying the distance of proxies towards the user's line of sight between 5°to 25°. A lower angular distance results in more clutter since all proxies are overlaying each other independently of the direction they are pointing to. A higher angular distance will result in worse perception of the shapes and a less accurate estimation of the position of out-of-view objects (cf., [12, 21]).

Guidance towards out-of-view objects. Since we implemented out-of-view visualization techniques, we assumed that no visualization is necessary when the objects are visible on screen. However during pilot tests, some participants stated that our artificial out-of-view objects all look the same and therefore, they were not able to decide which object they were searching for when multiple out-of-view objects were closer together. To solve this problem, we decided to have 5° at the border of the screen where the visualization technique remains active. However, it was still problematic for participants to distinguish between objects close together. This resulted in 14 wrongly selected objects in VR and 14 wrongly selected objects in AR. As an outcome of this, we recommend not hiding the visualization when it is unclear which object the user is searching for, or to replace it with an on-display proxy.

Visualization technique performance. Both experiments in VR and AR showed that our techniques are perceived as usable for searching and selecting out-of-view objects. This is further supported by an average search time of 2.25 seconds in VR and 3.92 seconds in AR. However, our quantitative results showed that *HaloAR* performs significantly better with regard to direction error than *WedgeAR* (*HaloAR*=29.79°, *WedgeAR*=36.03°). Nevertheless, the correlation between the direction towards out-of-view objects and the direction error is stronger for VR (*HaloVR*=0.663, *WedgeVR*=0.710) than for AR (*HaloAR*=0.294, *WedgeAR*=0.372), which indicates that users were better at estimating smaller angles towards out-of-view objects in VR than in AR. To avoid different graphic performances between the platforms, we reduced the amount of polygons rendered to a level that is supported by the Hololens and used the same amount of polygons also for the Oculus.

LIMITATIONS

Reduced 3D perception. Since our visualizations (per individual device) are always positioned with the same distance to the user's line of sight, it is not possible to look at the 3D proxy from different angles, making it harder perceive the full volumetric shape of the proxies. This limitation is part of our approach to position our proxies at the same position in the user's periphery and not in the environment. However, by attaching the visualizations to the head of the user, we can ensure that the proxies are always visible and do not go out-of-view.

Ecological validity. We tested both visualization techniques in VR as well as in AR in a controlled lab study. To measure user performance under these settings, we needed to control the environment as much as possible. However, this limits our understanding of how such techniques can be used across real applications and use-cases (e.g., representing objects / characters in VR games). Nevertheless, our work invites such ecological testing of out-of-view visualization techniques as a future research agenda.

CONCLUSION

In this paper, we developed *HaloVR*, *WedgeVR*, *HaloAR* and *WedgeAR* to visualize the position of out-of-view objects. Our findings showed that all techniques are usable under constraints of their specific technology (VR or AR). We showed two different approaches of using our techniques (foveal visualization and peripheral visualization). We found that the limited FOV in current AR devices has a negative impact on user performance. Our work opens up avenues for further investigating out of view object visualization techniques, where we believe ecological testing and lowering direction estimation error will improve the adoption of such visualization approaches.

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:` **`http://dx.doi.org/10.1145/642611.642695`**
2. John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
3. 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:` **`http://dx.doi.org/10.1145/1152215.1152266`**
4. 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:` **`http://dx.doi.org/10.1145/1456650.1456652`**
5. Uwe Gruenefeld, Abdallah El Ali, Wilko Heuten, and Susanne Boll. 2017a. 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:` **`http://dx.doi.org/10.1145/3098279.3122124`**
6. Uwe Gruenefeld, Dag Ennenga, Abdallah El Ali, Wilko Heuten, and Susanne Boll. 2017b. 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)*. ACM, New York, NY, USA, 109–118. `DOI:` **`http://dx.doi.org/10.1145/3131277.3132175`**
7. 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:` **`http://dx.doi.org/10.1145/1357054.1357179`**
8. 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:` **`http://dx.doi.org/10.1145/1240866.1241014`**
9. 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:` **`http://dx.doi.org/10.1016/S0166-4115(08)62386-9`**
10. Niels Henze and Susanne Boll. 2010. Evaluation of an Off-screen Visualization for Magic Lens and Dynamic Peephole Interfaces. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 191–194. `DOI:`**`http://dx.doi.org/10.1145/1851600.1851632`**
11. 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:`**`http://dx.doi.org/10.1145/1868914.1869002`**
12. James Kalat. 2015. *Biological psychology*. Nelson Education.

13. Yen-Chen Lin, Yung-Ju Chang, Hou-Ning Hu, Hsien-Tzu Cheng, Chi-Wen Huang, and Min Sun. 2017. Tell Me Where to Look: Investigating Ways for Assisting Focus in 360 Video. In *Proc. CHI '17*. ACM, New York, NY, USA, 2535–2545. DOI: http://dx.doi.org/10.1145/3025453.3025757

14. 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: http://dx.doi.org/10.1145/2858036.2858339

15. Albert Michotte, Georges Thines, George. Butterworth, and Alan. Costall. 1991. *Michotte's experimental phenomenology of perception / edited by Georges Thines, Alan Costall, George Butterworth*. L. Erlbaum Associates Hillsdale, N.J. xi, 258 p. : pages.

16. Takuro Nakuo and Kai Kunze. 2016. Smart Glasses with a Peripheral Vision Display. In *Proc. UbiComp '16*. ACM, New York, NY, USA, 341–344. DOI: http://dx.doi.org/10.1145/2968219.2971393

17. Leif Oppermann, Lisa Blum, and Marius Shekow. 2016. Playing on AREEF: Evaluation of an Underwater Augmented Reality Game for Kids. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 330–340. DOI:http://dx.doi.org/10.1145/2935334.2935368

18. P. Perea, D. Morand, and L. Nigay. 2017. [POSTER] Halo3D: A Technique for Visualizing Off-Screen Points of Interest in Mobile Augmented Reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 170–175. DOI: http://dx.doi.org/10.1109/ISMAR-Adjunct.2017.58

19. 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: http://dx.doi.org/10.1145/1753326.1753581

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

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

22. Robert Xiao and Hrvoje Benko. 2016. Augmenting the Field-of-View of Head-Mounted Displays with Sparse Peripheral Displays. In *Proc. CHI '16*. ACM, New York, NY, USA, 1221–1232. DOI: http://dx.doi.org/10.1145/2858036.2858212

23. 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:http://dx.doi.org/10.1145/765891.766022