FlyingARrow: Pointing Towards Out-of-View Objects on Augmented Reality Devices

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

Daniel Lange
University of Oldenburg
Oldenburg, Germany
daniel.lange@uol.de

Lasse Hammer
University of Oldenburg
Oldenburg, Germany
lasse.hammer@uol.de

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

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

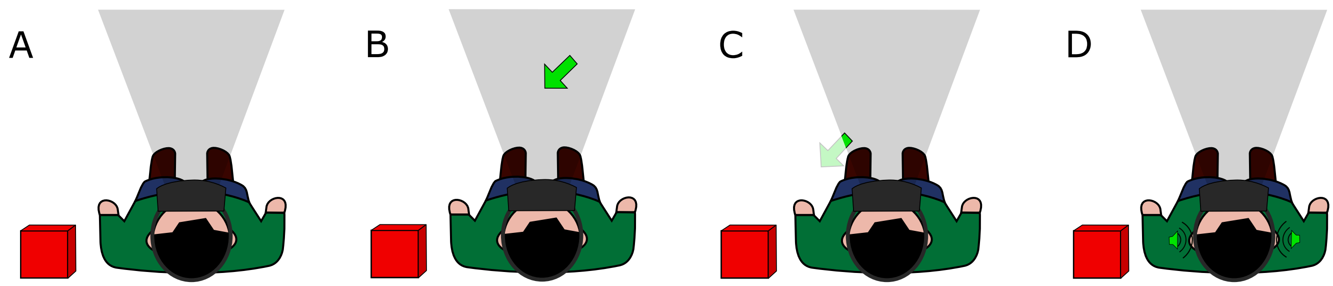


Figure 1: **FlyingARrow (green) as image sequence (A-D), out-of-view object (red), field of view (gray).** ***Best seen in color.***

ABSTRACT

Augmented Reality (AR) devices empower users to enrich their surroundings by pinning digital content onto real world objects. However, current AR devices suffer from having small fields of view, making the process of locating spatially distributed digital content similar to looking through a keyhole. Previous solutions are not suitable to address the problem of locating digital content out of view on small field of view devices because of visual clutter. Therefore, we developed *FlyingARrow*, which consists of a visual representation that flies on-demand from the user's line of sight toward the position of the out-of-view object and returns an acoustic signal over headphones if reached. We compared our technique with the out-of-view object visualization technique *EyeSee360* and found that it resulted in higher usability and lower workload. However, *FlyingARrow* performed slightly worse with respect to search time and direction error. Furthermore, we discuss the challenges, and opportunities in combining visual and acoustic representations to overcome visual clutter.

PerDis '18, June 6–8, 2018, Munich, Germany

ISBN 978-1-4503-5765-4/18/06...$15.00

DOI: https://doi.org/10.1145/3205873.3205881

ACM Classification Keywords

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

Author Keywords

Arrow; head-mounted; augmented reality; hololens; out-of-view; off-screen; visualization technique

INTRODUCTION

Over the past few years, head-mounted Augmented Reality (AR) devices have steadily been gaining in popularity. Their potential was shown by early products, specifically smart glasses (e.g., Google Glass[1]). These glasses are capable of overlaying the real world with digital information (e.g., for navigation tasks [16]). However, more recent products such as the Microsoft HoloLens offer more pervasive augmented reality by allowing users to pin digital content onto real world objects. Therefore, they allow digital content to merge with real objects, without having to switch between them. Building such devices is technologically challenging, which explains why all current existing head-mounted AR devices suffer from having too small fields of view. For comparision, the field of view of the HoloLens is more than ten times smaller than the human field of view [19]. Therefore, digital content placed in the existing environment is frequently hidden out-of-view.

[1] https://de.wikipedia.org/wiki/Google_Glass, last retrieved April 11, 2018

Localizing such out-of-view digital content thus becomes challenging for users (e.g., when these objects move or user's do not remember where they placed them). A recent approach to provide a solution to this problem was *EyeSee360* , which uses radar-like visualization to show the positions of out-of-view objects. However, *EyeSee360* has a high workload and adds clutter for devices with smaller fields of view [7]. Therefore, the problem of perceiving single out-of-view objects on small field-of-view AR devices remains unsolved.

In our approach, we combine the visual representation of an out-of-view object with an acoustic signal to point towards these objects in 3D space. As shown by previous work, this combination leads to faster reaction times [13]. Further, the combination of both modalities helps to reduce the visual information load [18] and clutter on screen. We use an 3D arrow that flies from the user's line of sight towards the position of the out-of-view object and returns an acoustic signal on it. Whereby, the audio signal is not used to encode the 3D position of the object instead it indicates that the 3D arrow reached the out-of-view object.

In this paper, we ask: **(RQ1) How fast can out-of-view objects be found, (RQ2) how accurately can their positions be estimated and (RQ3) how usable and workload-intense is the experience of using our FlyingARrow technique compared to EyeSee360 in Augmented Reality?** To evaluate the performance of *FlyingARrow*, we conducted a user study in AR using the Microsoft HoloLens.

Our research contributions are:

- A multi-modal visualization technique to point to out-of-view objects on small screen devices.
- Comparison of *FlyingARrow* to a the radar-like visualization technique *EyeSee360*.

RELATED WORK

In this section, we examine existing work divided into three areas. We start with discussing off-screen visualization techniques followed by audio(-visual) representations and the reasoning for choosing sound as second modality. We conclude with recent work on out-of-view visualization techniques and explain why we chose *EyeSee360* to compare with.

Pointing towards off-screen objects

Since our work is inspired by an existing off-screen visualization technique, we cover a subset of research in that space. The three main approaches proposed in prior work for the visualization of off-screen objects on small displays include: Overview+detail, Contextual views, and Focus+context [8, 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 their transitions between focus and context. In the Focus+Context approach the transition is soft (e.g., fisheye-views that convey a distorted view [17]), while 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 study them in more detail.

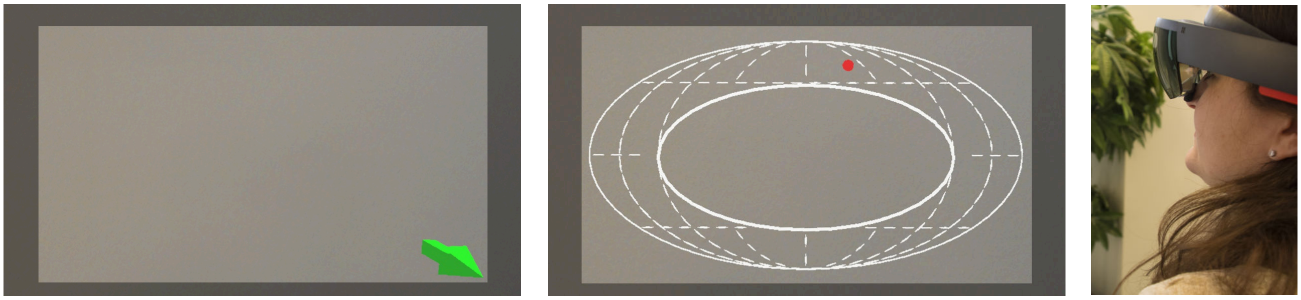
One of the first Contextual views was presented by Zellweger et al. [20]. Although they provided contextual information along the borders, users found it difficult to guess the actual positions of the off-screen objects. Therefore, Halo was suggested as an improvement [1]. It uses circles drawn with their centers around the off-screen objects and cuts 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 shaped arrows are used to point toward off-screen objects. Several studies compared Halo with Arrow approaches [3, 10], revealing that Arrows with fixed sizes performed worse than Halo, while scaled arrows performed slightly better. The amount of visible objects also has a high impact on performance.

Techniques based on audio(-visual) representations

Since our approach utilizes both visual and audio representations, we cover a small subset of research using these representations. In the work 'Head-Mounted Sensory Augmentation Device: Comparing Haptic and Audio Modality' the authors Kerdegarhi et al. showed that audio feedback is useful for aiding in indoor navigation [11]. However, the used cues were not multimodal and therefore, lacked the visual part. Loecken et al. showed that participants could react significantly faster to light cues when audio cues were added [13]. In our approach, we chose audio as our second modality because it is already integrated in all head-mounted Virtual and Augmented Reality devices, and therefore requires no additional effort to be implemented. Further, previous work showed it lead to higher reaction times when added to existing visual stimuli.

Out-of-view visualization techniques

Lin et al. [12] 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 was already gone or a part of a scene that already took place (e.g., a tackling in soccer). This highlights the need for accurate visualization of out-of-view objects. In our recent work [6], we adapted Arrow, Halo and Wedge to head-mounted Augmented Reality. Our results showed that all of these techniques are applicable to 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 [7]. EyeSee360 is explained in section 4. However, EyeSee360 has a high workload and adds clutter to devices with smaller fields of view. We chose Arrow as an inspiration for our technique because it uses smaller representations than Halo and Wedge. Further, it is the only approach that allows to use it's representation as an object that flies towards out-of-view objects and therefore, to test whether the idea of amodal completion can be transferred to uniform movement.



(a) *FlyingARrow* on HoloLens. (b) EyeSee360 on HoloLens. (c) Participant during study.

Figure 2: **FlyingARrow and EyeSee360 on field of view of HoloLens.** ***Best seen in color.***

GENERAL APPROACH

We explore how to guide users toward out-of-view objects in the 360° around the user. Our technique *FlyingARrow* uses a combination of visual and auditive representation to point toward out-of-view objects. For guiding toward out-of-view objects the 3D distance between the out-of-view object and the user is not relevant. Therefore, we stick with encoding the 3D directions toward these objects. Our work draws on our previous work [6], which showed that out-of-view object visualization can be successfully adapted from 2D off-screen visualization techniques to head-mounted Augmented Reality. In an user study we compare how well our technique *FlyingARrow* performs in comparison to *EyeSee360*.

EYESEE360

EyeSee360 [7] is a visualization technique that allows a user to know the position of out-of-view objects. Figure 2b shows how *EyeSee360* looks like on the HoloLens. This grid system compresses 3D position information onto a single 2D plane. The inner ellipse of *EyeSee360* represents the FOV of the current user, and the area outside the ellipse is outside the user's view. Each dotted line represents a 45° section of the user's view. The horizontal line expresses the altitude of the object. The vertical curved lines represent the horizontal direction of the object. For example, the red dot at the upper right part of *EyeSee360* represents an object that is almost 45° to the right and more than 45° up.

THE FLYINGARROW

Our developed technique *FlyingARrow* combines the visual representation of an out-of-view object with an acoustic signal to point toward the object in 3D space. The combination of both modalities helps to reduce the visual information load [18] and clutter on the screen. Similar to *amodal completion* or *amodal perception* [14, 5], where users can mentally complete simple shapes even when only part of the shape is visible, we hypothesize that users can mentally complete an uniform movement when only one part of that movement is visible. Therefore, we use a 3D arrow (cf. Figure 2a) that flies with uniform movement from the user's line of sight toward the position of the out-of-view object and returns an acoustic signal on it (see Figure 1). We hypothesize that the user can mentally complete the movement out of view. We decided to use another modality to inform the user when the 3D-Arrow reaches the out-of-view object. We chose sound because all current devices are equipped with audio and because previous work showed that visual(audio) cues are useful for reducing reaction times [13]. The 3D direction of the out-of-view object is encoded by the direction in which the 3D-Arrow is pointing, and the distance is encoded by the flight duration of the arrow. We implemented *FlyingARrow* in the 3D game engine Unity[2]. It will be available as an Open Source project on GitHub[3] and will support various AR as well as VR devices (e.g., Microsoft HoloLens or Oculus Rift).

Identifying parameters of FlyingARrow

We identified various parameters to adjust our technique *FlyingARrow* to small field of view devices. Therefore, we derived fitting settings of these parameters from related work or with pilot testing.

Speed of 3D-Arrow is connected to flight duration because there is a limited time period in which humans can memorize perceptions. This period is stated by prior work as being up to three seconds (also called the three-seconds-phenomenon) [15]. Therefore, we adapt the speed of the 3D-Arrow so that it is able to fly from the user's line of sight to the out-of-view object within three seconds.

Size of 3D-Arrow is chosen based on findings in pilot testing that investigated the smallest arrow size for which users still could perceive the shape and the direction. Since the field-of-view of the HoloLens is rather small, the 3D Arrow had to use at least one-twentieth of the screen space to be easily perceivable by the participants (cp. Figure 2a).

Sound variants are useful with regard to the amount of information that can be encoded. From simple sound coming from both speakers we could advance to 3D sound coming from the direction of the out-of-view object. Further, we thought of using periodic repeated sounds at one second intervals to assist in mentally understanding trajectory. However, since none of these worked best during pilot testing, we used the simplest possible sound pattern in which a single sound is given on the out-of-view object from both speakers, without information.

[2] http://www.unity3d.com, last retrieved April 11, 2018
[3] https://github.com/UweGruenefeld/OutOfView

EXPERIMENT WITH THE HOLOLENS

Study Design

To evaluate the performance of our novel visualization technique *FlyingARrow*, we conducted a within-subjects controlled laboratory study in Augmented Reality with the Microsoft Hololens. Our study's only independent variable was technique with two levels (*FlyingARrow* vs. EyeSee360). We used quantitative methods to evaluate user performance taking search time, search error and direction error as our dependent variables. Search time is measured as the time a user needs to locate and select an out-of-view object in the scene, while search error is specified as the number of objects a user wrongly selected. 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 it's correct position in 3D space.

For this study, we asked: **Does the visualization technique FlyingARrow perform better than EyeSee360 on a small field-of-view Augmented Reality device with respect to search time and search error (RQ1), direction accuracy (RQ2), perceived usability, and workload (RQ3)?**

H_1 We expect FlyingARrow to result in lower search time than EyeSee360.

H_2 Based on previous work, we hypothesize that uniform movement used in FlyingARrow can be completed in a mentally similar way to amdoal completion, and therefore lead to better direction estimation accuracy than EyeSee360.

H_3 We expect FlyingARrow to be less workload intense compared to EyeSee360.

Procedure

The within subjects study was divided into two tasks: a search task and a direction estimation task. Both tasks were divided into two blocks, with each block testing one technique (FlyingARrow, EyeSee360). We counter-balanced the two blocks across all participants. The out-of-view objects were randomly distributed in 3D space. We stored the seeds of the position generation to test the same positions for each technique. However, by randomly picking the order, we ensured that participants would not recognize a previous pattern of positions from the foregoing technique.

Task A: Search time

Each block of this task started with three test trials (not included in results), along with an explanation of the visualization technique and the task to achieve. In each run of this task, the participant had to find the out-of-view object with the support of the technique by selecting it with a cursor and a remote control. Each block was tested 10 times.

Task B: Direction estimation

Each block of this task started with three test trials (not included in results) and an explanation of the visualization and the task to achieve. In each run of this task, the participant had to estimate the position of a randomly placed out-of-view object. Each block was tested 10 times.

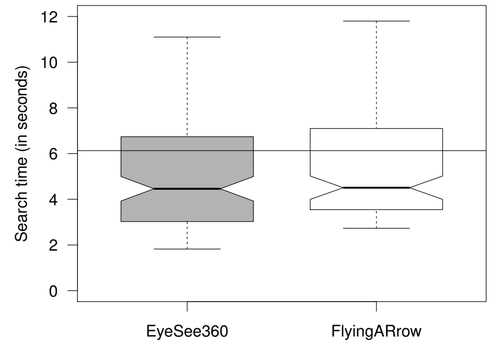


Figure 3: **Comparison of search times of visualization techniques (line indicates mean search time).**

After all blocks, we asked participants to fill out an SUS questionnaire and a RAW-TLX questionnaire for each technique. Further, participants were asked to fill out our individual subjective questionnaire and a demographic questionnaire. Overall, each participant took approximately 40 minutes to finish the experiment.

Participants

We recruited 12 participants (5 female), aged between 20 and 54 (M=27, SD=8.96). None of them suffered from color vision impairment. All had normal or corrected to normal vision.

Results

Search task

For the search task, we consider the effects of one factor (Visualization) 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: *FlyingARrow*=6.24s and *EyeSee360*=6.01s. The total number of wrongly selected objects are: *FlyingARrow* (40/120 = 33.3% search error) and *EyeSee360* (19/120 = 15.9% search error). The search times are compared in Figure 3.

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 ($W = 3312$, $Z = -0.832$, $p = 0.407$, $\phi = 0.05$). This indicates that *FlyingARrow* and *EyeSee360* do not significantly differ with respect to search time.

Estimation task

We consider the effects of one factor (Visualization) on mean direction error. The mean errors for the visualization techniques are: *FlyingARrow*=33.52° and *EyeSee360*=27.28°. The direction errors are compared in Figure 4. 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 ($W = 2972$, $Z = -1.723$, $p = 0.085$, $\phi = 0.11$). This indicates

Figure 4: **Comparison of direction error of visualization techniques (line indicates mean direction error).**

that *FlyingARrow* and *EyeSee360* do not significantly differ with respect to estimation accuracy.

RAW-TLX

For NASA Raw-TLX [9] scores, *FlyingARrow* scored 39.24 and *EyeSee360* scored 46.74. Both values indicate an acceptable workload, though *FlyingARrow* has a slightly lower workload than *EyeSee360*. A t-test revealed no significant difference between the *FlyingARrow* (M=39.24, SD=13.23) and *EyeSee360* (M=46.76, SD=10.52) conditions; t(22)=1.646, p=0.114.

System Usability Scale

For SUS scores, *FlyingARrow* scored 68 and *EyeSee360* scored 51. Therefore, *FlyingARrow* is over the threshold for acceptable usability while *EyeSee360* is not [2]. This shows that *FlyingARrow* is usable in AR.

Questionnaire

At the end of the study, we asked participants to answer four questions with 5-point Likert items. Participants stated that they were able to easily find the out-of-view objects with *FlyingARrow* (Md=4, IQR=1.25), while they were neutral for *EyeSee360* (Md=3, IQR=1). Furthermore, they stated that they were able to correctly estimate the position of out-of-view objects with *FlyingARrow* (Md=4, IQR=1.25), but not with *EyeSee360* (Md=2.5, IQR=1.25). Overall, seven participants preferred *FlyingARrow* while five preferred *EyeSee360*.

DISCUSSION

Advantages of head-mounted devices. Since our technique is inspired by off-screen visualization techniques [1, 8], it can be perceived as similar and therefore familiar to users. Especially, the simply shape of our 3D-Arrow is easy to understand. Combined with an head-mounted device, our technique can provide on-demand guidance for foveal presentation under more limited FOVs. Showing the visualization technique on-demand is especially helpful to avoid clutter on small field of view devices (e.g., HoloLens).

FlyingARrow vs. EyeSee360 Our main goal was to improve usability and reduce workload for visualization techniques on small field of view devices. Compared to *EyeSee360*, we reached that goal. However, *FlyingARrow* resulted in decreased performance in terms of search time, object selection accuracy, and direction estimation. We believe this is mostly due to lack of understanding of mentally complete uniform movement and may be improved in future work. However, a SUS-score of 68 for FlyingARrow is not indicating very good usability. We argue that this score is influenced by the HoloLens device. Three participants stated that the HoloLens itself felt uncomfortable to wear and that it negatively influenced their rating of both techniques.

Amodal completion for position estimation Besides the comparison to *EyeSee360*, our user study showed that users were able to estimate the direction of out-of-view objects or locate them in 3D space. From this, we can assume that mentally complete uniform movement works. However, future work is needed to improve the technique.

Multimodal technique In this paper, we showed that pointing to out-of-view objects can be done by splitting to multiple modalities. However, we still want to test different modalities such as tactile feedback, and we want to investigate redundant encoding of the 3D positions of out-of-view objects.

Limitations For each out-of-view object, our technique used a proxy flying from the user's line of sight toward the out-of-view object, but only once. Therefore, it was difficult for participants to locate an out-of-view object when the proxy was already gone. In order to overcome this limitation, we suggest either letting 3D-Arrows repeatedly fly toward the out-of-view object, or preventing 3D-Arrows from becoming out-of-view by sticking them at the screen border. Further, future work is needed to evaluate the performance of *FlyingARrow* in more realistic scenarios (e.g., gaming). We imagine that *FlyingARrow* may retain advantages over *EyeSee360* because of its reduced visual clutter.

CONCLUSION

In this paper we compared our novel visualization technique *FlyingARrow* with the previous technique *EyeSee360*. We showed that perceived usability and workload was lower for our technique *FlyingARrow*. Additionally, it reduced the amount of clutter added to the user' s screen. However, *EyeSee360* objectively performed best with regard to direction estimation, object selection accuracy and search time. However, we showed the potential of *FlyingARrow* and mentally complete uniform movement. Future work is required to further explore the combination of different modalities to increase the performance of direction estimation and search time for pointing toward out-of-view objects.

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. James Elder and Steven Zucker. 1993. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision Research* 33, 7 (1993), 981 – 991. DOI:http://dx.doi.org/https://doi.org/10.1016/0042-6989(93)90080-G
6. 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
7. 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
8. 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
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, 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
11. Hamideh Kerdegari, Yeongmi Kim, and Tony J. Prescott. 2016. Head-Mounted Sensory Augmentation Device: Comparing Haptic and Audio Modality. In *Biomimetic and Biohybrid Systems*, Nathan F. Lepora, Anna Mura, Michael Mangan, Paul F.M.J. Verschure, Marc Desmulliez, and Tony J. Prescott (Eds.). Springer International Publishing, Cham, 107–118.
12. 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
13. Andreas Löcken, Sarah Blum, Tim Claudius Stratmann, Uwe Gruenefeld, Wilko Heuten, Susanne Boll, and Steven van de Par. 2017. Effects of Location and Fade-in Time of (Audio-)Visual Cues on Response Times and Success-rates in a Dual-task Experiment. In *Proceedings of the ACM Symposium on Applied Perception (SAP '17)*. ACM, New York, NY, USA, Article 12, 4 pages. DOI: http://dx.doi.org/10.1145/3119881.3119894
14. 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.
15. Ernst Pöppel. 2002. *Three Seconds: A Temporal Platform for Conscious Activities*. Springer Berlin Heidelberg, Berlin, Heidelberg, 73–79. DOI: http://dx.doi.org/10.1007/978-3-642-50023-7_5
16. U. Rehman and S. Cao. 2017. Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE Transactions on Human-Machine Systems* 47, 1 (Feb 2017), 140–151. DOI:http://dx.doi.org/10.1109/THMS.2016.2620106
17. Manojit Sarkar and Marc H Brown. 1994. Graphical fisheye views. *Commun. ACM* 37, 12 (1994), 73–83.
18. Christopher D. Wickens. 2008. Multiple Resources and Mental Workload. *Human Factors* 50, 3 (2008), 449–455. DOI:http://dx.doi.org/10.1518/001872008X288394 PMID: 18689052.
19. 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
20. 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