User-Aware Rendering: Merging the Strengths of Device- and User-Perspective Rendering in Handheld AR

SEBASTIAN HUEBER, RWTH Aachen University, Germany
JOHANNES WILHELM, RWTH Aachen University, Germany
RENÉ SCHÄFER, RWTH Aachen University, Germany
SIMON VOELKER, RWTH Aachen University, Germany
JAN BORCHERS, RWTH Aachen University, Germany

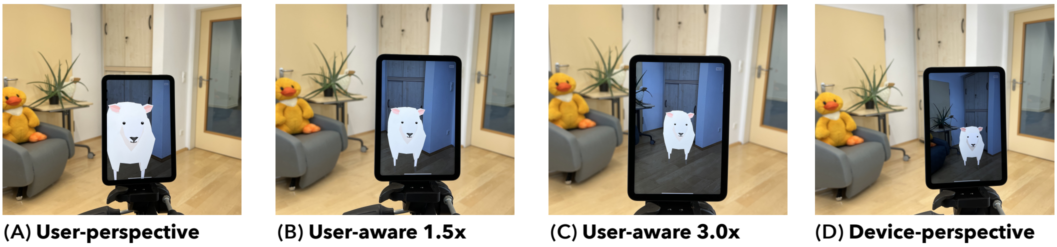

(A) User-perspective **(B) User-aware 1.5x** **(C) User-aware 3.0x** **(D) Device-perspective**

Fig. 1. Different rendering techniques when holding the device at an angle. User-Perspective Rendering (UPR, A) and Device-Perspective Rendering (DPR, D) differ in both the orientation from which the camera looks at the scene and their field of view (FOV). In UPR (A), the device aims for virtual transparency: the cupboard in the background is aligned between device viewport and peripheral vision. However, the FOV is limited and the sheep is slightly too large to fit on the screen. In contrast, DPR (D) creates a noticeable offset between screen and real world. For instance, the cupboard's real world and on-screen locations are disjoint in DPR. Our User-Aware Rendering (UAR) techniques (B, C) serve as a middle ground between the two, combining a large FOV with approximate alignment.

In handheld AR, users have only a small screen to see the augmented scene, making decisions about scene layout and rendering techniques crucial. Traditional device-perspective rendering (DPR) uses the device camera's full field of view, enabling fast scene exploration, but ignoring what the user sees around the device screen. In contrast, user-perspective rendering (UPR) emulates the feeling of looking through the device like a glass pane, which enhances depth perception, but severely limits the field of view in which virtual objects are displayed, impeding scene exploration and search.

We introduce the notion of User-Aware Rendering. By following the principles of UPR, but pretending the device is larger than it actually is, it combines the strengths of UPR and DPR. We present two studies showing that User-Aware AR imitating a 50% larger device successfully achieves both enhanced depth perception and fast scene exploration in typical search and selection tasks.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**.

Authors' addresses: Sebastian Hueber, RWTH Aachen University, Aachen, Germany, hueber@cs.rwth-aachen.de; Johannes Wilhelm, RWTH Aachen University, Aachen, Germany, johannes.wilhelm@rwth-aachen.de; René Schäfer, RWTH Aachen University, Aachen, Germany, rschaefer@cs.rwth-aachen.de; Simon Voelker, RWTH Aachen University, Aachen, Germany, voelker@cs.rwth-aachen.de; Jan Borchers, RWTH Aachen University, Aachen, Germany, borchers@cs.rwth-aachen.de.

2573-0142/2023/9-ART231 $15.00
https://doi.org/10.1145/3604278

Additional Key Words and Phrases: handheld augmented reality, field of view, head tracking, user-aware AR, user-perspective rendering, depth perception, selection times

ACM Reference Format:
Sebastian Hueber, Johannes Wilhelm, René Schäfer, Simon Voelker, and Jan Borchers. 2023. User-Aware Rendering: Merging the Strengths of Device- and User-Perspective Rendering in Handheld AR. *Proc. ACM Hum.-Comput. Interact.* 7, MHCI, Article 231 (September 2023), 18 pages. https://doi.org/10.1145/3604278

1 INTRODUCTION

Augmented Reality (AR) renders 3D objects into a view of the real world. The popularity of smartphones and their technical advancements have established handheld AR as the type of AR most commonly used by the masses [7]. Depth perception in handheld AR, however, is severely limited [14, 17, 19], leading to interaction problems when trying to navigate a scene or select targets. One limiting factor is that smartphones lack a stereoscopic image. Furthermore, depth perception is also limited because the image you see on screen uses *device-perspective rendering (DPR)*, i.e., only the position and orientation of the phone camera determine what is visible, as known from taking photos. This means that the device ignores your own field of vision, and moving your head around in front of your smartphone, as we naturally do to look at a scene from different angles, will not change what is displayed on screen.

User-perspective rendering (UPR) tries to overcome these issues by matching the *frustum*, a cut-off pyramid representing the field of view displayed on screen, to the area that the screen itself covers within the user's natural visual field [2, 21]. This offers a simple metaphor to the user: The device becomes a transparent glass window into the world that adds augmentation inside that window. This means that the frustum of UPR is dynamic. When holding the device closer to the face, it covers a larger part of the user's natural visual field, and thus the frustum extents increase. However, with typical distances between face and device of 40 cm [5], this results in a very narrow field of view (FOV), so that users can only see augmentations in a small part of the world around them: through a window the size of their device at their arm's length.

There are other limitations of user-perspective rendering: For a correct rendering, the camera frustum needs to originate in the user's eyes. However, without a stereoscopic screen, only one eye can be addressed precisely. This already results in a trade-off between using the device with one eye shut or accepting that horizontal alignment of the content is slightly off.

Yet even when these limitations of UPR are overcome, the increased realism obtained from it may not offer enough benefit for users to prefer it over the traditional, much wider-angle FOV of typical smartphone camera lenses [3]—simply because, with UPR, notably less content fits onto the screen while operating with a normal posture. If the FOV is too small, it becomes impossible to fit large scenes on one screen, which may also negatively affect depth perception[15] and search times [17, 25].

User-perspective rendering has been analyzed in many environments [2, 21, 33, 36], and the default device-perspective AR has reached the daily lives of many. However, *hybrid* rendering techniques that combine aspects of both aforementioned techniques have received no attention, opening up an interesting research opportunity: Is there a hybrid rendering approach that combines the strengths of DPR, like fast scene exploration, with the strengths of UPR, like enhanced depth perception? How do these different perspectives affect the AR experience and interaction? In this paper, we introduce *User-Aware Rendering* (UAR) as a novel approach: By taking a UPR implementation and virtually increasing the device size, we create a technique with a larger FOV that still reacts to the user's natural gaze direction. Moreover, this larger FOV also makes horizontal misalignments of the screen content less obvious and thus mitigates typical instabilities occurring in UPR systems.

In summary, the key contributions of our paper are:

- We introduce User-Aware Rendering as a new rendering and interaction technique for handheld AR.
- We present results from a study on how this technique affects depth perception, a measure in which UPR is known to perform better than DPR.
- We report how User-Aware Rendering affects search+select tasks, in which, thanks to its large FOV, DPR outperforms UPR.

In all, we are able to demonstrate that handheld AR leveraging User-Aware Rendering combines the strengths of UPR and DPR in these two areas, making it a promising candidate for many typical AR use cases.

In the remainder of this paper, we first review related work, then introduce User-Aware Rendering from a technical point of view before discussing our two user studies on depth perception and object selection. We close with limitations of our approach and resulting opportunities for further research.

2 RELATED WORK

In handheld AR, users see the virtual world through a single small screen. While using the system, people move their device around to change what is on screen. By doing so, they perceive 3D contents on a 2D screen by generating kinetic depth cues, such as the motion parallax induced through camera movement [17]. Motion parallax has long been known to have a strong impact on depth perception independent of other visual characteristics of the screen [11, 26]. With UPR, the on-screen content is dependent on the user's head as well. Thus, even without physical arm movements, users constantly trigger new inputs to the system, thus increasing motion parallax. In this section, we first present related work using UPR, and continue with the influence of visual characteristics on depth perception and the impact of the FOV on 3D scene understanding.

2.1 User-perspective Rendering

User-perspective rendering allows for AR "magic lenses" in their original sense as envisioned by Bier et al. [4] in the 90s: Instead of looking at the virtual scene through the perspective of the device, the device itself becomes transparent, like a sheet of glass. Objects appear in the same size and location in the vicinity of the user as they would if they were real, and thus, the mapping between augmented and real world is enhanced [33].

However, user-perspective rendering is a technically and computationally challenging problem. It requires tracking of the user's eyes and knowledge of the device location in the real world to calculate an off-axis projection of the virtual scene. Therefore, small errors in head tracking can diminish the effect. Prototype systems are not completely stable yet, require specialized hardware, and/or force the user into a specific, fixed position in front of the AR system [1]. One of the first handheld systems exploring head-coupled perspective rendering was pCubee by Stavness et al., a small fishtank VR with displays on each side of a cube [29]. Shortly after, handheld AR prototypes leveraging UPR were created with a variety of technical caveats. For instance, a system by Hill et al. required a fisheye camera and a fixed point of view from which the user had to observe the scene [13]. However, one can also compensate the fisheye camera with a homography transformation of the planar camera image to approximate UPR with fidelity [28, 32]. Baričević et al. were able to create a UPR simulation with acceptable stability with the integration of a stereo matching algorithm [2]. Mohr et al. increased the stability (though not correctness) of their UPR simulation by lowering the sampling rate of head input [21]. A main challenge across all prototypes remains that robust pixel-perfect alignment of the virtual scene has been impossible so far [1, 2, 21, 28].

While user-perspective AR could also allow for different interactions than device-perspective AR, this has received little attention so far. One example of such interaction techniques is using the user-perspective occlusion of the real world as a target selection mechanism [24].

Since making the device transparent massively narrows the FOV when holding it at arm's length, several studies favor the use of large screens when using user-perspective rendering [3, 22]. For example, Baričević et al. prototyped user-perspective AR with tablets and smartphones in VR. Their results of a search and select task show that user-perspective rendering could slightly enhance selection times when using a tablet. But more importantly, their study participants had a strong preference for device-perspective rendering when using a smartphone, as it allowed them to see much more of the scene at once by providing a significantly larger FOV [3].

2.2 Depth Perception

As handheld AR uses a single screen, depth perception cannot rely on physiological cues like stereoscopy to convey depth. Therefore, depth is inferred by our brain from pictorial and kinetic depth cues instead [6, 10, 12]. While UPR can enhance motion parallax, and thus kinetic depth cues, *pictorial depth cues* can be generated by the AR system independent of the chosen perspective. Pictorial depth cues include visual characteristics of the virtual objects, e.g., shading, shadows, relative size and shape of an object, or its texture [17]. While each of these various features enhances the visual realism of an object, neither increases depth perception significantly on its own. Instead, it is an interaction effect between all of them that enhances depth perception [8].

The virtual content itself also has an impact on perception, and the presence of unique landmarks and their shape can also support spatial tasks [8, 27]. The complexity of the virtual content's shape, its color, and texture luminance also impact depth perception [9].

Likewise, light and shadow can help users perceive depth. Highlights that are manipulated with the viewport can enhance the depth effect, e.g., in UPR a light could be placed at the user's head position [20, 36]. Much depth information is inferred from a 2D representation of the scene, as can be seen in shadow projections. Therefore, drop shadows perform better than ray-traced shadows even though they look less realistic [8].

Still, even with a realistic AR rendering pipeline, the limited depth cues current systems provide are insufficient to judge the distance from user to virtual content precisely, whether using a smartphone or tablet as viewport [14, 19]. This issue seems to be partially intrinsic to the content being virtual, independent of rendering quality [31]. Additional depth cues can be created by shading relative to a physical reference point, e.g., a secondary input device [35].

2.3 Impact of FOV

The size of the FOV determines how much of the virtual scene fits on the screen. Therefore, the difficulty of visual tasks increases with a limited FOV [17]. The FOV is also known to have a strong influence on distance perception. For example, when observing content through small FOVs, humans tend to overestimate distance as the content appears zoomed in. Large FOVs have the contrary effect [15]. When searching for annotations in large virtual models, a larger FOV leads to faster completion times [25]. While handheld AR uses central vision, wide-FOV AR systems are also possible. In these, people will take longer to notice changes in their peripheral vision area [30].

Common mobile AR using DPR suffers from the *dual-view problem*, a term used to summarize the three mismatches of the on-screen content with the real world: Different FOV, non-centered screen capture, and an angular offset of views [33]. All of them are present in Fig. 1, where a comparison between UPR (A) and DPR (D) shows that the virtual content has a different on-screen size and a different alignment with the real world. Especially the viewing angle offset can bias inter-object relations of the virtual content [17]. Čopič Pucihar et al. showed that users of UPR have a higher spatial perception in comparison to DPR [33].

3 USER-AWARE AR

By mitigating the dual-view problem and offering increased motion parallax, UPR seems to be a promising technique for enhancing AR experiences. However, even small errors or jitters in the head tracking can diminish the entire effect. Especially the need to“zoom in” the camera image in order to align the content makes tracking errors easily noticeable.

3.1 Concept

The motivation for our user-aware rendering (UAR) was to combine the advantages of UPR and DPR without increased hardware requirements. We knew from the related work that a small FOV can lead to perceptual issues and makes any tracking errors more noticeable. Therefore, UAR was designed to mitigate only two out of three aspects of the dual-view problem, focusing on the alignment of the overall content while remaining flexible in content size. In Fig. 1(A–C), the center of the on-screen content spatially correctly overlaps its real-world counterpart using both UPR and UAR. The increased content scales of UAR, however, render content smaller so that more fits on screen. This results in a slight misalignment towards the edge of the screen which increases with higher content scale factors. Due to the device borders separating the screen from the real world, however, especially with UAR 1.5x this offset is very minor. Using DPR, however, the position of on-screen content is defined by the device position only. Fig. 1(D) shows how the center location of the on-screen content does not match its real world counterpart and content at the edge of the screen is noticeably apart from its real-world counterpart.

Computationally, what we see through a virtual camera is defined by two matrices. The transformation matrix defines the camera location and orientation in space. The projection matrix defines the visible frustum in front of the clipping plane (i.e., screen). Figure 2 provides a visualization of different frustums.

When using a device-perspective frustum, the “eye” that observes the scene is the device camera (Fig. 2). Thus, the transformation matrix encodes the location and orientation of the device, and the projection matrix is a constant matrix that fits to the device camera’s characteristics. This stands in contrast to the user having her own visual field and the device only covering a small part of this area. For instance, in Fig. 1D, there is only a limited spatial relationship between the background image on the device vs. around the device: Although the device only covers the lower part of the cupboard, the entire cupboard is visible on the screen.

User-perspective rendering overcomes this issue by calculating a dynamic frustum that converges at the user, depicting the parts of the scene that are covered by the device. Thus, in Fig. 1A, only the lower part of the cupboard can be seen, and the camera image is approximately aligned with the real world. Since smartphone displays are not stereoscopic, one has to define one location for the camera inside the user’s head: This can be the right or the left eye or the center between both eyes as an approximation.

3.2 Prototype System

UAR borrows from the calculation of a dynamic viewing frustum from UPR, yet it increases its FOV by pretending that the device used is actually larger (Fig. 2), i.e., its screen corners are farther away from the center of the screen. Therefore, we created an AR implementation that supports UPR with an adjustable device size parameter, as well as DPR. To implement our prototype, we used the Unity graphics engine and tracking functionalities in Apple’s ARKit, and created a custom UPR

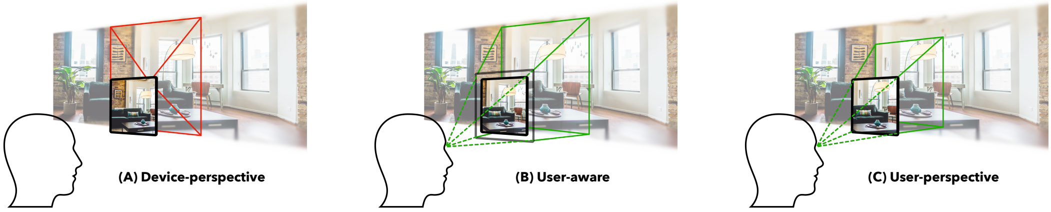


Fig. 2. Comparison of the camera frustum when the user is looking at the device at an angle. In DPR (A), the frustum of the virtual camera is completely defined by the hardware camera. In UPR (C), the virtual camera sits in the user's eye and is defined by the corners of the screen. Therefore, only a fraction of the actual camera image is visible. In UAR (B), we increase the size of the virtual screen (semitransparent frame around device), which lets the user see a larger part of the camera image.

rendering pipeline using techniques found in previous work. We adapted these implementations to support a variable device size. Tracking and rendering were performed at 60 Hz.

3.2.1 Camera Transform. The transformation matrix of the virtual camera can be derived directly from the head tracking capabilities of ARKit. When users have both eyes open, we place the camera at the center location between both eyes, creating an image that addresses our two-eyed vision. If the user closes one eye, the camera is placed into the open eye.

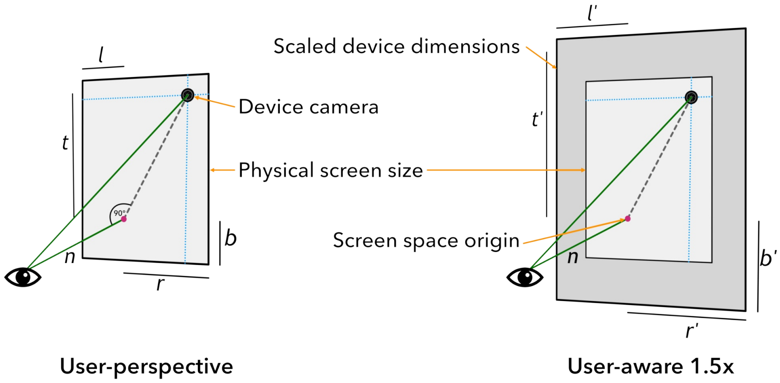
3.2.2 Camera Projection. The projection matrix P can be built from six parameters: the z distances of the near and far clipping planes (n and f), and the frustum extents on the near plane (t, r, b, l) [16].

$$P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{1}$$

For UPR, the corners of the camera frustum need to be cast from the eye location through the corners of the screen. Geometrically, one can obtain these parameters as follows: n can be defined by the Euclidean distance of the vector along the normal of the screen between device and camera (eye) location. t, r, b, l need to be the horizontal / vertical distances from the screen space origin (under the tip of the frustum) to the edges of the screen (Fig. 3). Thus, in UPR $r - l$ is the physical width of the screen / clipping plane. ARKit provides the location of the hardware camera. One can infer the four coordinates of the screen corners by measuring the physical distances from the sensor to the individual edges of the screen in advance. This way, the output of the rendering algorithm shows exactly what is covered by the device in the vicinity of the user.

The goal of UAR is to provide a larger FOV. Thus, the coordinates of the screen edges used to calculate t, r, b, l are spaced further apart (Fig. 3). This is achieved by pretending we measured larger actual distances in the previous step.

3.2.3 Video Feed. Lastly, the camera image needs to be placed in the background of the virtual content. For DPR this is easy, as the projection is already suitable for the camera. For UPR this is harder, as the planar camera image has to be mapped behind the 3D content so that the anchoring required for AR holds. The back-facing camera of mobile devices offers developers planar images only, which, e.g., cannot be mapped to a skybox to be used as background for the virtual scene. We



User-perspective

User-aware 1.5x

Fig. 3. The screen space origin (pink dot on light gray plane) is the closest point on the screen plane (light gray) to the eye. It is easy to calculate, as both the location of eye and hardware camera are given, and the normal of the plane is known (green). As the distances between camera sensor and screen edges (blue dotted lines) are constant, one can easily infer the values for t, r, b, l. With UAR, the distances between camera sensor and screen edges are scaled up, and by doing so, result in larger values for t, r, b, l and ultimately a larger FOV.

used the approach of Samini and Palmerius and placed the camera image on a plane behind the virtual objects [28].

For our device, using their method resulted in an image plane that is 10 m × 13 m large and roughly 10 m away from the screen. However, to maintain correct alignment with the real world, the image plane needs to be positioned logarithmically further away from the device camera based on the camera intrinsics for scenes that are directly within the first 2 m behind the device.. To identify a suitable mapping, we manually adjusted the distance of the image plane d based on the scene depth s for 60 samples with a varying scene depth between 0.2 and 7.5 m and interpolated these data points.

$$d = 10 + 23.5e^{-1.85d} \tag{2}$$

In addition, users could look through the device at an angle. Therefore, the image plane needs to be rotated dynamically around the camera to counterbalance possible misalignments [28].

The horizontal rotation r_h of the image plane is calculated by using the following formula:

$$r_h = \alpha - tan^{-1}\left(\frac{0.5 \times w}{d} \times \frac{tan(\alpha)}{tan(0.5 \times HFOV)}\right), \tag{3}$$

where w is the width of the image plane, and α the horizontal angle between the screen normal and the line of sight of the user. The vertical rotation follows analogously.

The approach of positioning the image can be refined further; e.g., devices equipped with LIDAR scanners could segment the camera image into multiple layers [18]. But even without LIDAR, we can assume that the user visually focuses on the virtual content. Thus we can optimize the alignment of the camera image to the virtual content by using the distance to the object in the virtual scene as value for our scene depth s.

It is also possible that a user is facing a part of the scene to which no camera image exists as a background. In that case, we decided to show black color as it blended in with the front color of the device we used. This is a hardware limitation that could be mitigated with wider-angle cameras. Yet, as the typical FOV during operation is still smaller than in DPR, there is usually enough leeway to operate a UPR system without seeing these unspecified areas.

Overall, with this technique, we observed good alignment of real world, camera image, and virtual content for scenes that are at least 25 cm away from the device. When the next physical surface is closer than 25 cm, visual quality deteriorates due to different reasons. First, the image lacks resolution as we are digitally zooming into a small part of the camera feed. Second, the camera feed can also become grainy or blurry due to cropping the camera image into a peripheral and slightly unfocused area of the camera sensor. Third, towards the corners of the planar camera image the projection of real and virtual world might disperse.

3.3 Scaling Factors

Our idea of UAR was to extend the dynamic frustum of UPR that provides motion parallax with a larger FOV even when holding the device at arm's length. Therefore, our system applied a custom scale factor for the measured real world device size when calculating the frustums. Thus, the scale factor provides a continuous gauge to increase the FOV of the system. As one can see in Fig. 2, a UPR frustum is calculated by casting rays to the four corners of the screen. By virtually scaling up the device size, and thus relatively moving these points farther away from the center of the screen, the resulting FOV becomes bigger.

With mobile devices becoming larger and the expected uptake in foldables in the future, we picked an iPad mini with an 8.3" screen running iOS 15 for our user studies. This device was especially chosen for its ultra-wide front-facing camera that tracks the user. While most iPhone camera systems track the user's head fairly reliably within a cone of 30° for distances under 90 cm [34], the additional area is beneficial for fast physically interactive tasks like AR experiences.

As we were not sure which scaling factor would turn out most beneficial for AR interaction, we tested two different ones, which are also visible in Fig. 1. The rationale for these scaling factors was as follows:

The standard device-perspective AR is based on a wide-angle camera and offers roughly a 70° (vertical) FOV. UPR and UAR, however, have a dynamic FOV that changes based on the distance between face and device (smaller distance = wider FOV). Recently, Boccardo found the typical viewing distance from smartphones while standing to be 37.4 cm, slightly below the traditional near point for optometric examinations (40 cm) [5]. Our own preliminary observations confirm that we can assume a viewing distance of 40 cm. With the device size of an iPad mini, this results in a 23° FOV in *UPR* at this typical viewing distance. To examine the impact of the FOV in UAR, we selected two scaling factors: *UAR3.0x* denotes a 3.0x device magnification, resulting in an FOV of around 63°, and thus comparable to DPR. On the other hand, we also wanted to test a magnification level between *UPR* and *UAR3.0x*. *UAR1.5x* halves this magnification to a 1.5x magnification of the device. It results in a noticeably larger FOV than UPR, which is still only about half as large (35°) as the DPR FOV.

One beneficial side effect is that with UAR techniques, slight issues in alignment are no longer as noticeable as with UPR, mitigating the horizontal alignment problem in two-eyed UPR usage.

We conducted two user studies to find out how UAR performs compared to DPR and UPR and whether it is able to combine their individual strengths, not their weaknesses. Moreover, we analyzed the device and some body movements to identify whether people change their behavior while operating UAR in comparison to DPR and UPR AR.

Our research questions were:

RQ1: Can UAR convey cues that enhance depth perception similarly to UPR?
RQ2: Does the increased FOV of UAR simplify search and selection tasks compared to UPR?
RQ3: Does UAR impact the amount of body movement while using AR?

Click the buttons in the order the cubes appear from frontmost to backmost

Fig. 4. In study 1, participants observed a scene with three cubes and had to judge their apparent depth order.

4 STUDY 1: DEPTH PERCEPTION

To better understand how our user-aware rendering techniques impact depth perception in handheld AR and the physical effort required, we conducted a first user study. With our task, we aimed to evaluate to which extent the kinetic depth cues achieved from altering the camera frustum help to identify the order of virtual mid-air objects. To trace effects measured back to the visualization used, we removed pictorial depth cues like shadows and reflections. In this first user study, we had four conditions: We compared *UAR1.5x* and *UAR3.0x* with the two baselines (*UPR* and *DPR*).

12 participants aged from 23 to 29 took part in the study ($M = 25.9, SD = 3.13$), eight male and four female. Four reported no previous usage of AR or VR. The others reported that they use AR or VR systems only occasionally.

4.1 Apparatus and Task

We created a set of 18 different AR scenes that contained a constellation of three virtual cubes each. The cubes could appear in a space of $3.0{\times}2.5{\times}5.0\,\mathrm{m}^3$ positioned 2 m in front of the participant. The diameter of the objects in a scene differed between 0.2 m and 0.8 m, so that the relative size could not be used as a depth cue. All objects had at least 0.3 m distance to the ground, i.e., they were floating in mid-air so that the floor could not be used as a reference point. Due to the rendering and study setup, the depth cues our participants could use were the perspective projection, occlusions, and motion parallax from changing the mobile phone camera position (Fig. 4). Object colors were randomized while ensuring a similar contrast and color saturation ratio. We asked participants if they suffered from color blindness to select a suitable color scheme.

Participants were asked to stand at a certain predefined location, observe the AR scene through the device, and determine the distance of the three virtual cubes from front to back. Objects were identified by their horizontal position in the scene and their color. We asked our participants to be as precise as possible while not becoming unnecessarily slow. People were asked to hold the tablet comfortably with either hand or bimanually, and were encouraged to rest their upper arm on their body to prevent fatigue. We created consistent lighting conditions by activating ceiling lamps over the virtual scene. The iPad display was set to 80% brightness to ensure legibility without becoming too warm over the course of the study.

We used a within-subjects study design, i.e., all participants tested all four conditions. We counterbalanced their order using Latin squares, in which each condition will precede another condition exactly once. AR scene order was randomized in each condition. All conditions started with an opportunity for participants to familiarize themselves with the rendering technique and explore how the device reacted to their input. To allow for familiarization with ordinary 3D models without giving away information about the task, we created a scene with multiple barnyard animals

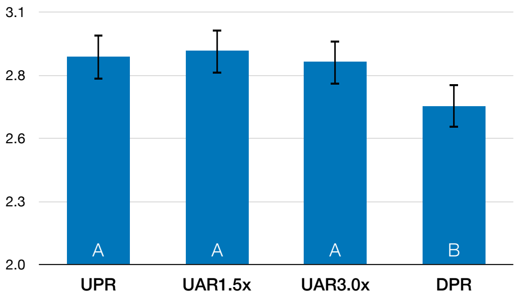


Fig. 5. Mean *Depth Scores* measured in the four conditions. *DPR* is the only technique that not reacts to head inputs. It resulted in significantly worse average depth perception than any other condition. Levels that do not share a letter (A, B) are significantly different (all $p < .05$). Whiskers denote 95% CI.

placed on the floor across the room. After each condition, we asked participants to rate their experience with this technique on 5-point Likert scales in a questionnaire. They had to rate which of the following four visual characteristics they found most helpful while solving the task: occlusion, motion parallax, anchoring of items in the real world, and grouping of virtual items. In addition, we asked them whether they found the task physically and mentally demanding, as well as whether the on-screen content was the one they intuitively expected, and whether they found the viewport to be restrictive, prohibiting them from seeing the whole scene comfortably. Each condition consisted of a set of 18 cube constellations that were tested in random order. Due to learning effects, we were not able to use the same set of cube constellations across all conditions despite randomization. Therefore, we used seven measured constellations that were the same across all conditions. The other 11 filler constellations were specified randomly. Only the data of the measured constellations was used for evaluation.

4.2 Variables

We used TECHNIQUE [*UPR*, *DPR*, *UAR1.5x*, *UAR3.0x*] as the main independent variable.

We measured *Depth Score* as the number of correct relative orderings in the participant's answer, with three being the highest and zero the lowest score possible. We calculated *Device Movement* as the traveled distance the device was moved while solving the task. We calculated *Head Movement* as the traveled distance the head was moved while solving the task relative to the tablet. We also measured the *Time* it took a participant to solve the task after a scene became visible.

4.3 Results

In this study, we were interested in the effect that user-aware rendering TECHNIQUES had on depth perception.

To analyze the effect of TECHNIQUE on our dependent variables, we used one-way ANOVAs for evaluation and Student's t-tests for post-hoc pairwise comparisons on the aggregated data for each participant and technique.

4.3.1 Depth Perception. *UAR1.5x* achieved the highest depth score (2.93), which corresponds to a correctness of 98%. Scores were similar in *UPR* (97%) and *UAR3.0x* (96%). The average success with *DPR* was only 90%. There was a significant main effect of TECHNIQUE on the *Depth Score* ($F(3, 33 = 3.430, p = .028)$. *DPR* performed significantly worse than *UAR1.5x* ($p = .007$), *UPR* ($p = .015$) and *UAR3.0x* ($p < .029$). Other comparisons were not significant. The individual scores per condition are visualized in Fig. 5.

4.3.2 Movement and Speed. We could not find a significant effect of the rendering technique on *Device Movement* ($F(3, 33) = 1.818, p = .163$). On average, our participants moved the device in similar amounts in all tested conditions. On average, we measured 2.5 m with *DPR*, 2.2 m with *UPR*, 2.1 m with *UAR1.5x*, and 2.0 m with *UAR3.0x*.

It also took our participants a similar *time* to solve the task across all conditions. We measured an average of 13.4 s with *UPR*, 11.7 s with *UAR1.5x*, 11.1 s with *DPR*, and 10.4 s with *UAR3.0x*. Thus, we noticed that the task was solved slightly slower in conditions with small FOVs. However, this effect was not significant ($F(3, 33) = 2.120, p = .117$).

4.3.3 Questionnaire Data and Comments. In the questionnaires, we asked participants to rate what features of the visualization they used to solve the task on a 5-point Likert scale (1 = completely agree). They reported a similarly high use of occlusion effects ($M = 1.3, SD = 0.7$) and motion parallax ($M = 2.0, SD = 1.0$) across all conditions. Reference points in the real world close to virtual objects ($M = 3.4, SD = 1.3$) were used less frequently, also because there was no furniture in the area in which objects could appear. We could not identify a significant effect between techniques using a Wilcoxon signed rank test.

When it comes to comments on the overall AR experience, our participants found that the size of the AR content prohibited them from seeing the whole scene in *UPR* ($M = 1.9$), a significantly worse experience than any of the other three conditions ($p < .02$). The intuitiveness of the visible camera frustum was not rated significantly different across conditions. Yet, *DPR* and *UAR1.5x* were rated slightly better ($M = 1.9$) than the other two ($M = 2.8$). This is partially because people know the behavior of *DPR* "from taking photos". The virtual transparency of *UPR* irritated one participant, who asked why the camera zoom changes when moving her head.

4.4 Discussion

The task in study 1 tested how the different rendering techniques affected depth perception. In this study, *UPR* performed significantly better than *DPR*, increasing the success in depth perception from 90% to 97%. The good performance of *UPR* was expected from the related work [15].

As *DPR* suffers from the-dual view problem, it varies from *UPR* in its FOV, the angular offset, and the center of the screen capture. Our *UAR* techniques all have different FOVs, yet they mitigate the other two aspects. Overall, we anticipated a decline in depth perception with a larger FOV, as this results in objects becoming smaller on screen, making depth cues harder to see. However, with *UAR3.0x*, we measured similar depth perception scores as with *UPR*. Both techniques provide motion parallax through a dynamic frustum.

Overall, the results were quite similar across all techniques. A possible reason for this is that our scenes often made it possible to achieve perspective overlapping between objects and thus being too simple. While using smaller objects could have increased the task difficulty, we still found *DPR* to perform significantly worse than the other techniques.

We used this setup to answer whether user-aware rendering is able to preserve this strength of UPR (RQ1). In our study, both *UAR1.5x* and *UAR3.0x* performed significantly better than *DPR* on average. Both UAR techniques were able to match the score of *UPR* (Fig. 5). This shows that the depth cues obtained from motion parallax were still usable independent of the increased FOV from our techniques.

DPR performed significantly worse than any other tested technique. The other three techniques leveraged continuous head tracking to change what is visible on screen, and thus resulted in a constant input stream of subtle motion parallax effects. While these effects are the likeliest explanation for this difference, it is important to note that motion parallax is also obtained in *DPR*

Fig. 6. The three phases of the task in study 2. Left: Participants had to tap on the red ball to spawn a target on the table. Middle: During the search phase, the frame was yellow as long as the target was not visible on screen. Right: When the target was in the viewport, the frame turned green, and participants tapped on the target to select it. The images depict DPR.

from moving around the device, and our participants all moved around a lot during operation; e.g., the tablet was moved 2.5 m on average in *DPR*.

Considering the similar time and movement across conditions and the questionnaire data and comments, this suggests that all rendering techniques were usable with ease and that participants did not have to adapt their usage patterns for our user-aware techniques. Overall, this study answered RQ1 positively.

5 STUDY 2: SEARCHING AND SELECTING OBJECTS

In many AR application domains, the virtual scenes can be too large to fit on one screen, e.g., when working with large data sets, virtual desktops, or reconstructions of buildings. This can complicate searching in these models [25]. To find out to which degree our user-aware techniques retain the fast scene exploration known from DPR, we conducted another study. This study was conducted after a 10 min break with the same group of people from study 1. Again, we used a within-groups design in which our participants tested the four conditions in a counterbalanced order, using a different Latin square than in study 1.

5.1 Apparatus and Task

We adapted the design of the search and selection task that Baričević et al. [3] used in VR for use in handheld AR. Participants were asked to stand in front of a 1.6×0.7 m table and select a virtual target on the screen by tapping on it. Targets could only appear in the mid-air space up to 0.5 m above the table.

To spawn a new target, participants had to tap a virtual ball to their right. As this ball was located below the table, it was intended to function as a homing target that shifted the AR frustum away from the table, so that targets needed to be searched for and were not immediately visible in front of the user. A border around the display indicated what to do next (Fig. 6): A red border implied that no object was visible and the homing target needed to be tapped. A yellow border was visible during the search phase when the object appeared over the table but was not visible in the current frustum. A green border denoted the selection phase, which was started once the object was visible on the screen for the first time. Our participants had to stand at most 1.2 m away from the table. Thus, even with DPR, only parts of the table fit onto the screen at once, as can be seen in Fig. 6.

The task was designed to minimize possible interaction effects of the task design by reducing the interaction to panning over a virtual scene and tapping an object. Participants were instructed to select the targets as fast as possible. We displayed their score on the top right screen corner to keep them engaged. For each condition we conducted three runs. The first one allowed participants

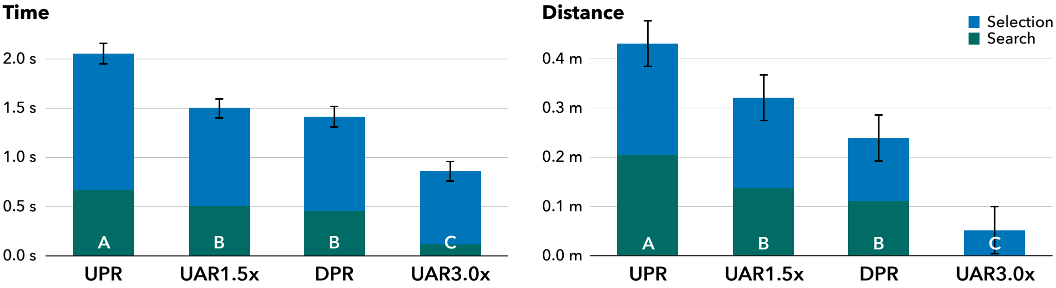


Fig. 7. Average time [s] (left) and device movement [m] (right) to search and select a target by condition. The two phases are color-coded into the graph. Using *UPR*, which offers the smallest FOV, our participants were the slowest and had to move the device the most. *UAR3.0x* required close to no search time because our participants held the device in such a way that the whole table fit on the screen. Levels that do not share a letter (A, B, C) are significantly different (all $p < .001$). Whiskers denote 95% CI.

to explore the technique and find out how to best operate the system. As suggested by Baričević et al. [3], this was limited to at most 8 min. Afterward, two runs taking 2 min each were made with a short break between them. We sampled 12 arrays containing 100 random target locations each once before conducting our studies. These 12 arrays were hard-coded in the software and used as the locations in the study for all participants in their individual runs. Thus all participants saw the same target locations in their n-th run. In a questionnaire after each condition, we asked them to rank on 5-point Likert scales which parts of their bodies they moved the most while solving the task. Options included tilting of head, hand, and movement of forearm, neck, and torso. We also asked them to briefly describe their search strategy in their own words.

5.2 Variables

We used TECHNIQUE [*UPR*, *DPR*, *UAR1.5x*, *UAR3.0x*] as the main independent variable. The two different PHASES [*Search*, *Selection*] were logged independently for additional analysis.

Time was measured for both phases. For the search phase, this is the time it took from tapping the homing ball till the target cube was rendered in AR on at least 1 px of the display. In the selection phase, it is the duration between the end of the search phase until the participant tapped on the cube. We calculated *Device Movement* as the traveled distance the device was moved (in m) while solving the task. Moreover, *Head Movement* denotes the traveled distance the head was moved (in m) while solving the task relative to the tablet.

5.3 Results

To analyze the effect of TECHNIQUE on our dependent variables, we used one-way ANOVAs for evaluation and Student's t-tests for post-hoc pairwise comparisons. Overall we obtained over 11,000 measures for analysis. To make the statistical test more reliable, we calculated the mean values for each participant, technique, phase and run. We then performed the analysis on the data averaged over both runs.

5.3.1 Time.

There was a significant effect of TECHNIQUE on *Time* ($F(3, 33) = 82.268, p < .0001$). *UPR* was slowest. Using this technique, it took our participants 2.05 s on average to select a cube after it appeared in the scene. With similar average times of 1.50 s and 1.41 s, *UAR1.5x* and *DPR* performed significantly better than *UPR* ($p < .0001$), and not significantly different from each other. *UAR3.0x* was faster than any other technique ($p < .0001$). However, this measurement has to be interpreted carefully, as this is due to the annihilation of the search phase with *UAR3.0x*: With all other techniques, the average search duration was 0.54 s (green parts in Fig. 7). Using *UAR3.0x* this measurement dipped to 0.11 s. The reason for this is that over the course of the study all participants found a way to hold the device so that both the table and the homing target were visible at once, thanks to the dynamic frustum of UAR.

5.3.2 Movement. There was also a significant effect of TECHNIQUE on *Device Movement* ($F(3, 33) = 21.790, p < .0001$). Device movement helps us understand how bodily the search interaction was using each technique. *UPR*, which already took the longest, also required the most device motion: 46 cm on average. *UAR1.5x* (33 cm) and *DPR* (27 cm) required significantly less device movement ($p < .001$). As mentioned above, *UAR3.0x* was used rather statically (9 cm on average) and cannot be compared properly here. Fig. 7 shows mean device movements during both phases. TECHNIQUE only had a significant effect on *Head Movement* when comparing the static usage of *UAR3.0x* with the other conditions.

5.3.3 Questionnaire Data and Comments. As part of the task, participants went over the scene from right to left while searching for the target. During their first run with this technique, all participants found a way to hold the device with *UAR3.0x* in such a way that the majority of the scene and some part of the homing target was visible on screen. To do so, they decreased the distance between their eyes and device, enforcing ultra-wide FOVs up to 100°. Thus, "no search was required": Upon tapping the homing target, a cube appeared on the screen. Its appearance was a strong visual cue to directly spot its location. Again, our participants described *UPR* as too restrictive in its narrow viewport.

As the interaction with *UAR3.0x* was not representative of the intended usage (see below), we left it out when analysing the ranking of body parts involved in controlling the AR system. Independent of the other three techniques, people preferred to rotate with torso movement ($M = 1.6, SD = 0.2$) and use their forearm ($M = 2.5, SD = 0.2$) to further adjust the frustum. Device tilt, head, and neck ($M = 3.6, SD = 0.2$) were not controlled actively but "rather subconsciously".

5.4 Discussion

Based on the visual characteristics of the different techniques, we already expected *DPR* to outperform *UPR* in this task. When combining search and selection time, UPR (2.0 s) took 43% longer than DPR (1.4 s) on average. This difference is due to the different visual output of these rendering techniques. First, the FOV of UPR is the smallest across all tested techniques. Secondly, the part of the virtual scene visible on screen can easily be influenced by rotating the device in DPR, but it only changes slightly in UPR. Instead, in UPR, the angle from which one is facing the device makes a difference. This difference in usage pattern is also visible when looking at the traveled distance while solving the task: UPR required more movement as the search was complicated with less FOV, and (in-place) rotations have no effect on what is visible through the screen.

When adding *UAR1.5x* and *UAR3.0x* into the mix, we see that both search and selection times, as well as device movements, decreased significantly the larger the FOV of the AR system was. Especially notable in the data is the non-existing search phase with *UAR3.0x* as a result of the large FOV. However, these measures are incomparable to the other techniques, as all participants found a way to abuse *UAR3.0x*. In order to solve the task faster by reducing movement required, they moved the device into a position so close to their face that most of the table and some part of the homing target were visible on screen. By doing so, upon tapping the homing icon, the target directly appeared inside the visible part of the scene. The device movement during the selection phase was then likely performed to better reach the target with the thumb for selection.

While this is an interesting strategy for special cases, regarding RQ2 it makes more sense to compare *UAR1.5x* with the two baselines. With a combined search and selection time of 1.5 s, this hybrid technique performed similar in speed to *DPR*. Significance tests prove that both were significantly faster than *UPR*. This is especially interesting as the foundation of user-aware rendering is in *UPR*: While one might expect that losing the ability of *DPR* to quickly pan over the scene by rotating the device in place might have a negative impact on performance, this was not the case. Looking at the graphs, one can rather see a relationship between FOV and search and selection times. This makes it likely that user-aware AR with a device scale factor between our two versions could yield a result even better than *DPR*.

Just like in study 1, the effect size between the best and worst performing techniques is rather subtle. Search and selection using *UAR1.5x* took only 0.5 s less than *UPR*, which still is a speed increase of 25%. However, one also needs to consider that our task design was very simplistic. As AR tracking and rendering is usually provided by a system library and not required to be implemented by app developers, future versions of AR toolkits could leverage head input for enhanced AR perception without effort required by app developers.

In our two studies, UAR was able to match the performance of the respective existing favorable rendering technique [3, 14, 17, 19, 25]. The design of both studies was reduced to very concrete aspects of perception. We saw in study 1 that a constant input stream of subtle motion parallax effects enhances depth perception. In study 2 we measured the positive impact of a larger FOV on scene exploration. Real-world AR experiences require both of these aspects: Both estimating the distance and order of multiple objects and getting an overview on the overall constellation of objects is relevant for any AR experience. Thus overall, UAR provides an interesting technique that combines motion parallax with a large FOV. Both the comments our participants made and the quantitative measurements suggest that the FOV of around 35° one obtains with *UAR1.5x* at a typical usage distance between device and head offers a sweet spot at which users can see enough content on screen (Study 2) while still benefiting from the added motion parallax for depth perception (Study 1). Thus, we attribute the enhanced performance measured with UAR to the more useful FOV it provides at a comfortable viewing distance.

6 SUMMARY AND CONCLUSION

We introduced *User-Aware Rendering*, a new approach to handheld augmented reality. It combines the better visual context and depth perception of user-perspective AR with the larger FOV of traditional device-perspective rendering that enables fast scene exploration. The technique works by calculating UPR frustums with virtually increased device sizes. We tested two scaling factors, 1.5x and 3.0x.

Our two studies provide valuable insights into how people perceive content in handheld AR. Each study focused on an individual known strength of *DPR* and *UPR*. Study 1 suggests that the additional motion parallax effect obtained from head tracking had a positive effect on depth perception (RQ1). Study 2 shows us that a large FOV is also important to quickly search for an object in the virtual scene (RQ2). On the other hand, our participants could not use the angular offset of *DPR* to an advantage in decreasing search times. We have seen no change in head or device movement in study 1 while operating the techniques, showing that they allowed for natural usage (RQ3). In study 2, larger FOVs resulted in reduced movement to solve the task, proving that the idea to scale up the virtual device size made sense. Our user-aware techniques, especially *UAR1.5x*, were able to combine the strengths of *UPR* and *DPR* in these studies by mitigating the dual-view problem without lowering the FOV.

7 FUTURE WORK

Our studies show that hybrid rendering techniques like UAR can combine large FOVs with motion parallax from head input to enhance the overall AR experience. UAR, however, is not be the only imaginable implementation to achieve this combination. We can also envision a rendering technique that is based on DPR instead of UPR that pans into specific areas of the captured camera. This approach would result in a similar amount, yet a different type of motion parallax effect. Thus, it could lead to a performance similar to *UAR1.5x* while being less computationally complex.

Our UAR approach also tested only two possible scaling factors for user-aware rendering. Oftentimes in our studies, we found a relationship between FOV and performance. Altering this scale factor offers further interesting research trajectories. We can also envision a dynamic scale factor based on the visible virtual content, with the device zooming in to show smaller scenes while increasing the FOV for large scenes.

User-Aware Rendering should also be evaluated on other screen sizes. That said, in our preliminary tests, the prototype also worked well on a phone, and the impact of device size is likely small, as operating distance decreases with a smaller device [23]. Consequently, the relative size of the area blocked in the user's field of vision remains steady. This work identified that the FOV has a strong impact on how people perceive handheld AR. Thinking beyond mobile devices, it is likely that FOV manipulations could have an impact on HMDs as well.

Finally, the transformation of the background image based on the floor and model distance worked well for us. Still, there were minor issues in this perspective transformation, which could be refined by using a different (fish-eye) camera. None of our participants mentioned any alignment issues of the virtual content in the real world. However, the black borders around the camera image that were especially present when *UAR3.0x* ran out of camera feed should be tackled with different camera technology.

Overall, this work serves as a first exploration of our concept of User-Aware Rendering in handheld AR. Our two studies demonstrate its promise, as it was able to combine the advantages, but not the disadvantages, of UPR and DPR. We hope our contributions help other HCI researchers and practitioners further explore this exciting new approach to the fast-growing field of handheld AR.

ACKNOWLEDGMENTS

This work was partly funded by the German B-IT Foundation.

REFERENCES

[1] Daniel Andersen, Voicu Popescu, Chengyuan Lin, Maria Eugenia Cabrera, Aditya Shanghavi, and Juan Wachs. 2016. A Hand-Held, Self-Contained Simulated Transparent Display. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 96–101. https://doi.org/10.1109/ISMAR-Adjunct.2016.0049

[2] Domagoj Baričević, Tobias Höllerer, Pradeep Sen, and Matthew Turk. 2014. User-Perspective Augmented Reality Magic Lens from Gradients. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (Edinburgh, Scotland) *(VRST '14)*. Association for Computing Machinery, New York, NY, USA, 87–96. https://doi.org/10.1145/2671015.2671027

[3] Domagoj Baričević, Cha Lee, Matthew Turk, Tobias Hö, and Doug A. Bowman. 2012. A Hand-Held AR Magic Lens with User-Perspective Rendering. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 197–206. https://doi.org/10.1109/ISMAR.2012.6402557

[4] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. 1993. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) *(SIGGRAPH '93)*. Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/166117.166126

[5] Laura Boccardo. 2021. Viewing distance of smartphones in presbyopic and non-presbyopic age. *Journal of Optometry* 14, 2 (2021), 120–126. https://doi.org/10.1016/j.optom.2020.08.001

[6] James E. Cutting and Peter M. Vishton. 1995. Chapter 3 - Perceiving Layout and Knowing Distances: The Integration, Relative Potency, and Contextual Use of Different Information about Depth*. In *Perception of Space and Motion*, William Epstein and Sheena Rogers (Eds.). Academic Press, San Diego, 69–117. https://doi.org/10.1016/B978-012240530-3/50005-5

[7] Arindam Dey, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan. 2018. A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014. *Frontiers in Robotics and AI* 5 (2018), 37. https://doi.org/10.3389/frobt.2018.00037

[8] Catherine Diaz, Michael Walker, Danielle Albers Szafir, and Daniel Szafir. 2017. Designing for Depth Perceptions in Augmented Reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 111–122. https://doi.org/10.1109/ISMAR.2017.28

[9] Tiffany D. Do, Joseph J. LaViola, and Ryan P. McMahan. 2020. The Effects of Object Shape, Fidelity, Color, and Luminance on Depth Perception in Handheld Mobile Augmented Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 64–72. https://doi.org/10.1109/ISMAR50242.2020.00026

[10] David Drascic and Paul Milgram. 1996. Perceptual issues in augmented reality. In *Stereoscopic Displays and Virtual Reality Systems III*, Mark T. Bolas, Scott S. Fisher, Mark T. Bolas, Scott S. Fisher, and John O. Merritt (Eds.), Vol. 2653. International Society for Optics and Photonics, SPIE, 123–134. https://doi.org/10.1117/12.237425

[11] C. Furmanski, R. Azuma, and M. Daily. 2002. Augmented-reality visualizations guided by cognition: perceptual heuristics for combining visible and obscured information. In *Proceedings. International Symposium on Mixed and Augmented Reality*. 215–320. https://doi.org/10.1109/ISMAR.2002.1115091

[12] H. Hecht, R. Schwartz, and M. Atherton. 2003. Looking into pictures : an interdisciplinary approach to pictorial space.

[13] Alex Hill, Jacob Schiefer, Jeff Wilson, Brian Davidson, Maribeth Gandy, and Blair MacIntyre. 2011. Virtual transparency: Introducing parallax view into video see-through AR. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 239–240. https://doi.org/10.1109/ISMAR.2011.6092395

[14] J. Edward Swan II, Liisa Kuparinen, Scott Rapson, and Christian Sandor. 2017. Visually Perceived Distance Judgments: Tablet-Based Augmented Reality Versus the Real World. *International Journal of Human–Computer Interaction* 33, 7 (2017), 576–591. https://doi.org/10.1080/10447318.2016.1265783 arXiv:https://doi.org/10.1080/10447318.2016.1265783

[15] Paul B. Kline and Bob G. Witmer. 1996. Distance Perception in Virtual Environments: Effects of Field of View and Surface Texture at Near Distances. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 40, 22 (1996), 1112–1116. https://doi.org/10.1177/154193129604002201 arXiv:https://doi.org/10.1177/154193129604002201

[16] Robert Kooima. 2009. Generalized perspective projection. *J. Sch. Electron. Eng. Comput. Sci* 6 (2009).

[17] Ernst Kruijff, J. Edward Swan, and Steven Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. 3–12. https://doi.org/10.1109/ISMAR.2010.5643530

[18] Victor Kyriazakos and Konstantinos Moustakas. 2015. A User-Perspective View for Mobile AR Systems Using Discrete Depth Segmentation. In *2015 International Conference on Cyberworlds (CW)*. 69–72. https://doi.org/10.1109/CW.2015.67

[19] Jingjing May Liu, Gayathri Narasimham, Jeanine K. Stefanucci, Sarah Creem-Regehr, and Bobby Bodenheimer. 2020. Distance Perception in Modern Mobile Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 196–200. https://doi.org/10.1109/VRW50115.2020.00042

[20] Jack M. Loomis, Joshua M. Knapp, et al. 2003. Visual perception of egocentric distance in real and virtual environments. *Virtual and adaptive environments* 11 (2003), 21–46.

[21] Peter Mohr, Markus Tatzgern, Jens Grubert, Dieter Schmalstieg, and Denis Kalkofen. 2017. Adaptive user perspective rendering for Handheld Augmented Reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. 176–181. https://doi.org/10.1109/3DUI.2017.7893336

[22] Ji-young Oh and Hong Hua. 2006. User evaluations on form factors of tangible magic lenses. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. 23–32. https://doi.org/10.1109/ISMAR.2006.297790

[23] D Paillé. 2015. Impact of new digital technologies on posture. *Points de Vue* 72 (2015), 22–30.

[24] Yue Qin, Chun Yu, Wentao Yao, Jiachen Yao, Chen Liang, Yueting Weng, Yukang Yan, and Yuanchun Shi. 2023. Selecting Real-World Objects via User-Perspective Phone Occlusion. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 531, 13 pages. https://doi.org/10.1145/3544548.3580696

[25] Donghao Ren, Tibor Goldschwendt, YunSuk Chang, and Tobias Höllerer. 2016. Evaluating wide-field-of-view augmented reality with mixed reality simulation. In *2016 IEEE Virtual Reality (VR)*. 93–102. https://doi.org/10.1109/VR.2016.7504692

[26] Brian Rogers and Maureen Graham. 1979. Motion Parallax as an Independent Cue for Depth Perception. *Perception* 8, 2 (1979), 125–134. https://doi.org/10.1068/p080125

[27] Joan Sol Roo, Jean Basset, Pierre-Antoine Cinquin, and Martin Hachet. 2018. *Understanding Users' Capability to Transfer Information between Mixed and Virtual Reality: Position Estimation across Modalities and Perspectives.* Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173937

[28] Ali Samini and Karljohan Lundin Palmerius. 2014. A Perspective Geometry Approach to User-Perspective Rendering in Hand-Held Video See-through Augmented Reality. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (Edinburgh, Scotland) *(VRST '14)*. Association for Computing Machinery, New York, NY, USA, 207–208. https://doi.org/10.1145/2671015.2671127

[29] Ian Stavness, Billy Lam, and Sidney Fels. 2010. PCubee: A Perspective-Corrected Handheld Cubic Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1381–1390. https://doi.org/10.1145/1753326.1753535

[30] Xuetong Sun and Amitabh Varshney. 2018. Investigating Perception Time in the Far Peripheral Vision for Virtual and Augmented Reality. In *Proceedings of the 15th ACM Symposium on Applied Perception* (Vancouver, British Columbia, Canada) *(SAP '18)*. Association for Computing Machinery, New York, NY, USA, Article 13, 8 pages. https://doi.org/10.1145/3225153.3225160

[31] William B. Thompson, Peter Willemsen, Amy A. Gooch, Sarah H. Creem-Regehr, Jack M. Loomis, and Andrew C. Beall. 2004. Does the Quality of the Computer Graphics Matter when Judging Distances in Visually Immersive Environments? *Presence* 13, 5 (2004), 560–571. https://doi.org/10.1162/1054746042545292

[32] Makoto Tomioka, Sei Ikeda, and Kosuke Sato. 2013. Approximated user-perspective rendering in tablet-based augmented reality. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 21–28. https://doi.org/10.1109/ISMAR.2013.6671760

[33] Klen Čopič Pucihar, Paul Coulton, and Jason Alexander. 2013. Evaluating Dual-View Perceptual Issues in Handheld Augmented Reality: Device vs. User Perspective Rendering. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction* (Sydney, Australia) *(ICMI '13)*. Association for Computing Machinery, New York, NY, USA, 381–388. https://doi.org/10.1145/2522848.2522885

[34] Simon Voelker, Sebastian Hueber, Christian Holz, Christian Remy, and Nicolai Marquardt. 2020. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376578

[35] Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. 2020. Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376848

[36] Jing Yang, Shiheng Wang, and Gábor Sörös. 2018. User-Perspective Rendering for Handheld Applications. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 270–274. https://doi.org/10.1109/ISMAR-Adjunct.2018.00084

Received January 2023; revised May 2023; accepted June 2023