Effects of Hand Occlusion in Radial Mid-Air Menu Interaction in Augmented Reality

Nico Feld* Trier University
Daniel Zielasko† Trier University
Benjamin Weyers* Trier University

ABSTRACT

A major challenge in augmented reality applications is the occlusion of virtual objects by physical, real-world objects. Occlusion, however, is a major cue for perceiving the correct depth of objects and whether those are colliding or not. Obtaining an accurate depth perception is especially important for mid-air interactions since misperception can result in low usability. To investigate the effect of hand occlusion in radial mid-air menu interaction, we implemented a model-based method for hand occlusion and conducted a user study ($N = 30$) where basic mid-air interactions had to be performed with and without occlusion. Further, we were interested in investigating whether the effect of occlusion additionally depends on the interaction method used for the radial mid-air menu interaction, namely *pinching* or *tapping*. Contrary to our expectations, our results indicate both positive and negative effects occlusion for the interaction methods on usability. We identify and discuss possible side effects that could have led to these unexpected results.

Index Terms: Human-centered computing—HCI—Interaction paradigms—Mixed / augmented reality; Human-centered computing—HCI—Empirical studies in HCI;

1 INTRODUCTION

In augmented reality applications, accurate depth perception of real and virtual objects is essential for user interaction. Howard et al. [24] identified ten depth cues that lead to a correct depth estimation, which should ideally be simulated in AR for realistic depth perception. Some cues, like linear perspective and motion parallax, are typically generated by 3D engines in AR and virtual reality. However, the cues binocular convergence, accommodative focus, and occlusion are often neglected by Current AR systems due to technical limitations [40]. Recent research investigates solutions for these issues using new display technologies [9, 25] and techniques to implement occlusion [22, 37, 43, 20]. Occlusion is particularly significant for depth perception and user interaction, as it provides high-depth contrast regardless of distance by being an ordinal cue [14]. Thus, we assume that occlusion could significantly influence the usability of interaction methods that rely on correct depth perception. Therefore, we want to investigate the following research question: **Does hand occlusion improve interaction performance for standard interaction methods in AR?**

The interaction with mid-air menus, common in AR applications, is a use case that could benefit from occlusion. Mid-air menus are not only common for system control but also in the rising field of AR-IDEs for AR-assisted programming and debugging. These IDEs require a precise hand interaction to interact with the menus and with the source code itself for navigation [30, 31]. A low precision for these menus results in low usability and, thus, can lead to frustration. Hence, we choose mid-air menu interaction to explore the research question. Key interaction methods for mid-air menus include *tapping* and *pinching* [35, 34]. *Tapping* involves touching an object with a fingertip, often used in 2D interactions [11], while *pinching*, akin to a grabbing gesture, is common in 3D contexts [34]. As we can not exclude a potential interaction effect from the chosen interaction method on the impact of occlusion, we test both in separate conditions within an AR menu interaction tasks, each with and without hand occlusion.

*e-mail: {feldn, weyers}@uni-trier.de
†e-mail: daniel.zielasko@rwth-aachen.de

2 RELATED WORK

2.1 Implementation of Occlusion in AR

As previously stated, occlusion is a crucial depth cue in AR, with distinct methods to handle virtual and real object interactions [14]. While the occlusion within real or within virtual objects is trivial, combining both types, like a real object occluding a virtual object, is more complex [40]. Rendering virtual objects over real ones is straightforward, but optical see-through (OST) devices, like the Hololens 2, can not render virtual objects opaque enough to fully occlude the real object. Video see-through (VST) devices offer an alternative by rendering the real environment on a display, allowing full occlusion by virtual objects [40], but suffer from hardware limitations like high latency and low resolution. Hebborn et al. [22] outline three main methods for virtual object occlusion in AR for OST devices: model-based, object-based, and depth-based.

In the model-based method, a virtual copy of the real object is created for occlusion, requiring an accurate 3D model and pose estimation. The object-based approach uses the object's contour for occlusion, avoiding the need for a full 3D model but still requiring precise contour identification and pose estimation. Depth-based occlusion relies on depth maps from the environment, obtained via depth sensors [16, 32] or stereo vision [41], comparing these maps with virtual object depth for rendering decisions. This method does not need a 3D model or contour but requires additional device power and sensors to create these depth maps.

2.2 Hand interaction with occlusion

Recent advancements include integrating occlusion as a depth cue in AR for hand interaction, as demonstrated by Kim et al. [28], who developed a hybrid touch and hand gesture technique for handheld devices. In other AR implementations, like tabletop AR, virtual objects are positioned behind the user's body, avoiding direct occlusion. CAVE systems, though not typically classified as AR, face similar occlusion challenges, particularly with virtual objects needing to occlude physical hands [13]. These systems generally do not support occlusion of the real environment by virtual objects, unlike Head-Mounted Display (HMD) systems. As a result, in CAVE environments, interactions like the raycasting approach used by Gebhardt et al. [21] in pie menu navigation avoid direct hand interaction by maintaining a distance between menus and the user's hands.

Feng et al. [19] applied a model-based method for hand occlusion in AR, tracking the user's hands and overlaying a realistic hand model. When the user is grabbing a real tracked object that is substituted with a virtual object, e.g., changing the brand of a soda can, the virtual object would normally occlude the user's real hand. Their qualitative evaluation indicated that their approach enhances usefulness and usability compared to no occlusion.

Recently, Tang et al. [42] proposed a solution to occlusion in AR systems in their *GrabAR* prototype. In contrast to the work of Feng et al. [19], they did not replace real objects with virtual ones but placed virtual objects in the user's hand with occlusion. The user could then interact by using an object-based method. Their evaluation confirmed that the occlusion of the hand led to higher efficiency

979-8-3503-7449-0/24/$31.00 ©2024 IEEE
DOI 10.1109/VRW62533.2024.00106

and usability and resulted in significantly better results than other common techniques in generating correct occlusions.

2.3 Interaction methods: Tapping and Pinching

Feld & Weyers [18] implemented a 3D mid-air radial menu using a *pinching* method. In this menu, the user has to grab a floating orb with two fingers, drag it over a menu item, and then release the orb to select the corresponding item. While this mid-air radial menu is not at the center of their evaluation, the authors stated to us that their observations and the comments from the participants centered on two main aspects: First, many participants had difficulties estimating the correct depth of the mid-air radial menu and, therefore, often grabbed either in front of the orb or behind it. Second, they suggested changing the *pinching* method to a *tapping* method. These results both indicate that occlusion could improve usability and provide a promising menu design for both the *tapping* and the *pinching* method.

The *pinching* method, also known as "air-tap and hold" [12], is widely used for 3D manipulation in Hololens 2 applications [34]. The evaluation of the AR urban planner prototype by Buchmann et al. [8] highlights the intuitiveness of *pinching* for interaction. Buchmann et al. [8], the documentation of the Hololens 2 [34], and the Mixed Reality Toolkit (MRTK) [35] describe *pinching* as being a grabbing gesture, especially for small objects. According to Bowman [3], grabbing is a natural and intuitive interaction in 3D environments, often the first instinct for new AR users when interacting with virtual content. Therefore, several implementations even try to utilize grabbing for distant objects [4, 47]. The exploratory study by Piumsomboon et al. [39] about user-defined AR gestures indicates the popularity of *pinching* for 3D manipulation tasks. Moreover, when *pinching* is used for selection and dragging, like in the previous work by Feld & Weyers [18], it classifies as a "crossing-based interaction", possibly enhancing user experience in menu interactions [45].

In contrast to this physical-based method, *tapping* is a widespread interaction method for 2D virtual content [11]. We define *tapping* as touching an object with the fingertip, which also is called touching [28, 10, 7] or tipping [29, 33] in related work. This work uses *tapping* as a generic term for tapping, tipping, and touching. However, the *tapping* method is challenging in mid-air when tapping on virtual objects, as Chan et al. [10] showed in their work about implementing the *tapping* method for intangible displays. They described the same difficulties with the depth estimation as Feld & Weyers [18] experienced in their work but with the *tapping* method. To enhance the depth estimation, they implemented hand shadows and audio cues and found a significant improvement in the user's performance. While *tapping* is the standard interaction method in the MRTK by Microsoft for 2D surface interaction, it uses a proximity light [35] and a fingertip visualization [17] to enhance the depth perception for their mid-air menus. Bruder et al. [7] further observed reduced performance in mid-air touching on stereoscopic tabletop surfaces due to the vergence-accommodation conflict, blurring either the virtual object or the user's finger. Zielasko et al. [46] investigated the effect of passive haptics for both desk-aligned and mid-air menus using the *tapping* method. However, they found no significant effects of either passive haptics or menu alignment on performance and usability.

3 Hypotheses

Based on the related work, we formulate three hypotheses about the possible effects of the investigated factors hand occlusion and interaction methods. Given *pinching*'s relation to grabbing, it may be more intuitive to interact with 3D floating content and *tapping* with 2D surface-bound content. Thus, *pinching* could further benefit from introducing occlusion as it is performed in 3D and needs proper depth perception. *Tapping* is an inherently 2D interaction concept and, thus, may require less exact depth estimation. This could alter the effect of occlusion on usability depending on the interaction method used. Therefore, we expect an interaction effect between occlusion and the interaction method regarding usability, resulting in our first hypothesis: **(H1) There is an interaction effect between the factors interaction method (pinch, tap) and hand occlusion (on, off)**. However, because occlusion provides an additional depth cue for both interaction methods, we expect the interaction effect of our first hypothesis to be a non-cross-over interaction effect and a positive effect of occlusion regarding usability for both interaction methods: **(H2) Occlusion leads to higher usability for both interaction methods**, which splits into: **(H2.1) Occlusion leads to higher usability for pinching** and **(H2.2) Occlusion leads to higher usability for tapping**. Lastly, we assume *pinching* to be more intuitive in the context of 3D mid-air menu interaction, suggesting higher usability irrespective of occlusion: **(H3) The *pinching* method has higher usability than the *tapping* method**.

To investigate the hypotheses formulated in the previous Section, we first describe the implementation of our hand occlusion solution and a mid-air radial menu, which can be used with both interaction methods. We then used these implementations in the evaluation; see Section 5.

4 Implementation

4.1 Hand occlusion

In this work, we implemented a model-based method to create an artificial occlusion using a generated model of the users' hands, as explained in Section 2.1. This dynamic hand model is generated and provided by the MRTK during runtime via the hand-tracking capabilities of the Hololens 2. Figure **??** shows this implementation of the hand occlusion. By changing the model's color and shading to a solid black, the hand model gets rendered transparent for the user, as stated in the official documentation of the Hololens 2 [44]. In this case, if the hand model is occluding a virtual object in the scene, the occluded parts of the object are not being rendered, resulting in a perceived occlusion by the users' real hands. The major downside of this approach is the latency included in the rendering by the hand tracking, which is perceivable as slight asynchronicity of the occlusion with the user's hand movement. A camera-based video analysis yields an average "motion to photon" latency of $\sim 100ms$.

4.2 The Menu

For our study, we implemented a custom radial mid-air menu compatible with the Hololens 2 and hand tracking, inspired by the work from Feld & Weyers [18] (see Figure 1). Initially, the menu appears as a single orb in mid-air. Selecting this orb (via *pinching* or *tapping*) expands sub-menus circularly around it, forming a radial design. When selecting one of these sub-menus, additional items expand again. The various sub-menus and their items are all represented by a 3D object representing their functionality. For example, the object of a sub-menu for multiple colors could be an orb colored in rainbow colors, as done later in our evaluation in Section 5. If the user selects items of the sub-menu, the related functionality gets executed (see Section 5.3), and the whole menu collapses. In the used implementation, the menu has two levels. The menu has two levels: the first with a $15cm$ radius and the sub-menus with a $7cm$ radius. The expansion and collapse animations last $0.2s$ each. We gathered these parameters experimentally through testing. Both audio and visual feedback accompany selections, with visual feedback tailored to the interaction method. After the selection, the user can close the menu by selecting a confirmation object positioned under the main menu orb.

For the *tapping* method, the selection is made by a simple tap on the orb sub-menus or items. This selection (indication and confirmation), akin to intersecting the finger with the virtual object for selection, mirrors the evaluation of passive haptics for menus by Zielasko et al. [46]. To visually indicate a successful tap, the corresponding 3D objects move $2cm$ along the tap gesture, mimicking the movement of a traditional button.

With the *pinching* method, the user selects the orb by grabbing it with two fingers. When pinched, the menu expands into the sub-menus, maintaining its position relative to the initial position of the orb. The user can move the orb while keeping it pinched, and the menu remains fixed in position. To expand a sub-menu, the user moves the orb to a sub-menu orb without releasing the pinch. When the sub-menu is expanded, the user can move the orb to one of the sub-menu items. When the user releases the orb, the item gets selected, and its functionality gets executed. After selection, the

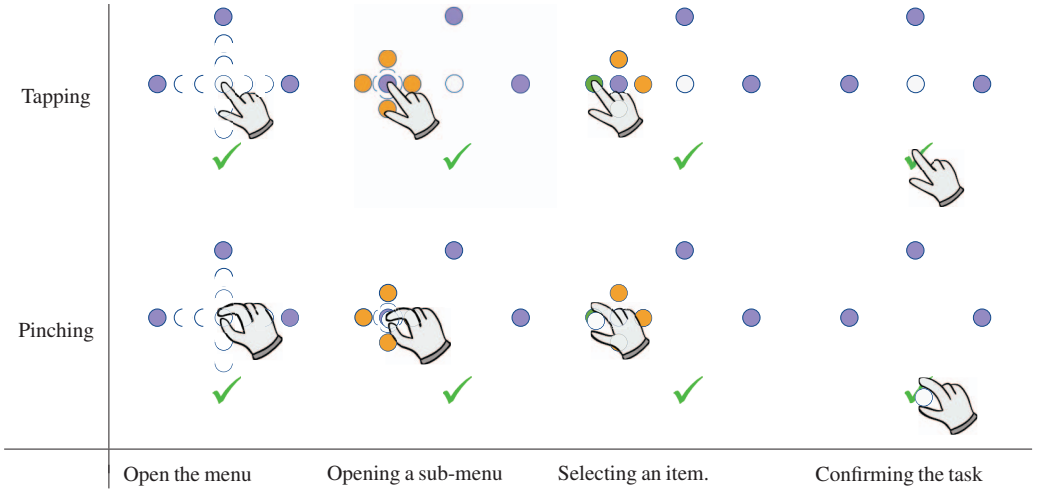

Figure 1: Menu design for the *tapping* and *pinching* method.

menu collapses, and the orb returns to its original position. Selecting the items only via the *pinching* gesture, therefore only clenching the fingers, without translating the orb, would counteract the intuitiveness of the *pinching* method, as it is a gesture to manipulate objects.

5 Evaluation

5.1 Study Design

We used a 2x2 within-subject study design, focusing on two factors: hand occlusion (on vs. off) and interaction method (*pinching* vs. *tapping*). This leads to the four conditions: **Pinch+** (*Pinching* with occlusion), **Pinch-** (*Pinching* without occlusion), **Tap+** (*Tapping* with occlusion), and **Tap-** (*Tapping* without occlusion). Our study design is inspired by the work of Zielasko et al. [46], who investigated the effect of passive haptic feedback on two menu types.

The study tasks comprise a naive search task, as defined by Bowman [3], for identifying an outstanding object among multiple other objects and a menu task that requires selecting the attributes of the outstanding object. The naive search task provides a natural context [46] of the menu task and rest periods to mitigate possible effects of arm fatigue [27]. Each participant repeated the combined task for all four conditions, resulting in a within-subject design.

To investigate all three hypotheses, a way to measure usability is required. The ISO 9241-11:2018 standard [26] describes that usability consists of effectiveness, efficiency, and satisfaction and is, for instance, measured by the System Usability Scale (SUS) questionnaire [6]. As the SUS yields only a subjective measure, we complemented the SUS with objective measures using two error rates (relating to effectiveness) and the measurement of the time-on-task (relating to efficiency).

5.2 Apparatus & Virtual Environment

The study was conducted in an empty room with a desk and seat for the participant. We allocated a space of approximately 3.5 *m* x 4 *m* for the study tasks (see Section 5.3). As an AR HMD, we used a Microsoft Hololens 2. The study task was implemented using Unity 2019.4 with MRTK 2.7.0.

In the center of the task space, a virtual white cylinder ($r = .4$ *m*, $h = .8$ *m*) was displayed .75 *m* above the ground, so the participants could move around the cylinder freely (see Figure 2). The cylinder's surface was equally covered with 100 objects, each 5x5x5 *cm* in size. All of the objects had the same, randomly chosen shape (capsule, cylinder, or cube), color (green, orange, purple, or white), and texture (circles, stars, stripes, or checkered), while one random object had a different shape, color, and texture to all others.

To display the menu, the participants had to tap their index fingers together. The menu appeared 30 *cm* from the participant towards the cylinder, positioned at a height between the waist and shoulders to minimize arm fatigue, as recommended by Hincapié-Ramos [23]. This height was determined using the distance of the Hololens 2 to the ground and average human body proportions from the DIN-33402-2 standard [15]. Ground detection was enabled by Vuforia image tracking, utilizing a floor-attached marker. Additionally, the menu was oriented around its local y-axis towards the user, ensuring ease of interaction.

5.3 Tasks

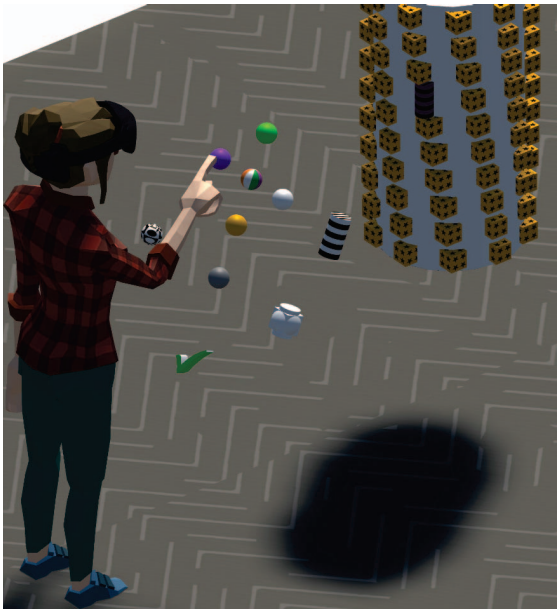

Figure 2: Sketch of the task design. The participant identified the outstanding object (purple, striped cylinder), already selected the shape and texture (indicated by the black and white striped cylinder), and is currently selecting the color with the *tapping* method.

As mentioned above, the task was subdivided into a naive search

task and a menu interaction task to increase external validity and potentially decrease fatigue. In the naive search task, participants identified an object on the cylinder differing in shape, color, and texture from others. Therefore, the participant could move around the cylinder freely by simply walking around it. We replaced the large cloud from Zielasko et al. [46] with a smaller cylinder, ensuring equal visibility of all objects without the need for manual checking. Furthermore, due to the cylinder's opaqueness, the participants could not see the objects on the other side of the cylinder and, thus, were forced to walk around the cylinder physically. This way, each participant had to move the same distance over each task and could not spot the object without moving. Additionally, our participants were forced to move physically, in contrast to the study by Zielasko et al. [46]. This required more space for walking, and thus, the limitation of the cylinder's size was necessary.

After participants identified the outstanding object, they had to tap their fingertips together to open the menu (see Section 5.2). Subsequently, they had to select all three attributes of the outstanding object using the menu. For this study, the sub-menus represent the shape, color, or texture attributes as selectable items representing the concrete attributes. Another floating object placed next to the menu previews the selected attributes to the study participant to provide feedback regarding the selected attributes. This object can be seen in Figure 2 as a white cylinder with black stripes, indicating that the participant already selected the texture (stripes) and the shape (cylinder) attribute. When the participants selected all attributes, they needed to confirm their selection by selecting a green checkmark in the menu. Following confirmation, the cylinder's objects changed attributes, a new unique object appeared, and the task restarted. To standardize the naive search task duration, the attribute order and unique objects were semi-randomly generated beforehand, ensuring each participant encountered the same scenarios so that the attributes did not repeat and were always different from the outstanding object.

5.4 Procedure

Initially, each participant signed an informed consent about collecting and using the data gathered and answered a few demographic questions. They were then assigned their first condition, with all conditions distributed using a balanced Latin square design. Each condition consisted of an introduction, the actual task, and filling out a questionnaire. The introduction included a video explaining the task and the menu. Afterward, they could familiarize themselves with the task and the menu for an unrestricted time. After the participants felt confident handling the menu and the task, they commenced the actual task, repeating the naive search and menu selection ten times per condition. Upon completion, they filled out a questionnaire regarding the current condition. After finishing all four conditions, participants were asked to complete a final questionnaire. The whole procedure took 45-60 minutes per participant and was approved by the university ethics board.

5.5 Participants

31 participants voluntarily took part in the study. Beforehand, a power analysis resulted in a minimum sample size of 20 participants, with an assumed medium effect size. The participants were recruited at the university campus and required to have no or a corrected visual impairment. For their participation, they were compensated with 10€/h. One of the participants was excluded from the analysis due to technical issues during the study, resulting in a total of 30 participants (13 female and 17 male, age $M = 25.63 \pm 3.499$). 13 (43%) reported prior experience with AR, 15 (50%) experience with 3D video games and 13 (43%) with 3DUIs, via simple yes/no questions. All participants, but one, stated to be right-handed.

5.6 Measures

Three objective measures were recorded during the experiment: *menu time*, *invalid interactions*, and *wrong confirms*. *Menu time* tracks the duration from initial menu interaction to attribute selection confirmation, indicating the efficiency of the menu. The *invalid interactions* measure is the ratio of invalid interactions (interactions made by the participant but not registered by the system) to all interactions (including the invalid interactions) per task. *Wrong confirms* is the total number of confirms when at least one attribute was incorrect. Thus, *invalid interactions* represents a technical error rate, possibly caused by wrong depth perception, and *wrong confirms* represents a task-related error rate, possibly caused by usability-related issues. As these measures characterize the fit of the operation to the actual task, enabling the user to work with the menu, they indicate the effectiveness of the menu. The *invalid interactions* were measured by capturing video footage of the participant's interaction with the menu and subsequent video analysis to quantify incorrect interactions in a second step. *Invalid interactions* were quantified via video analysis, with data from 28 of 30 participants due to recording consent. As ten tasks per condition were conducted, the measurements of these tasks were averaged, resulting in a single data point for each condition per measure.

Subjective measures included perceived usability and arm fatigue. Usability was assessed using the "System Usability Scale" (SUS) questionnaire [6], with a higher SUS score indicating better usability. Furthermore, arm fatigue was measured using the Borg questionnaire [2], where a higher score indicates greater fatigue. Participants also ranked the four conditions by fun and efficiency and rated the subjective impact of the occlusion latency on a 5-point Likert scale, informed by a latency definition. Finally, task difficulty was rated on a 5-point Likert scale, and additional comments about the study were collected.

6 Results

In the following, we summarize the results of our analysis. The complete inferential statistics are listed in table 1 and are reported as significant at a significance level of $\alpha = .05$. The descriptive statistics are depicted in Figure 3. The raw data can be found in the supplement material.

For the analysis of H1, both scaled and ordinal variables are analyzed with a two-way ANOVA with repeated measures for multiple factors. For the post-hoc tests, we apply Dunn tests with Bonferonni corrections. Ordinal variables with multiple factors are usually analyzed with a Friedman test, but in contrast to an ANOVA, it does not allow for an analysis of an interaction effect. However, according to Norman [36], an ANOVA can also be used for ordinal variables to analyze an interaction effect. For H2 and H3, the scaled variables are analyzed with t-tests and the ordinal variables with Wilcoxon tests. The rankings for efficiency and fun are analyzed using Friedman tests.

Regarding **H1**, we expect an interaction effect between hand occlusion and the interaction method, but only the ANOVA of the *SUS-Score* reveals a significant interaction effect ($p = .007$). To allow for more insights into the found effects of **H1**, we exploratory investigate the main effects of the found interaction effect of the *SUS-Score*. While we find no main effect of hand occlusion, we find a main effect of the interaction method for *menu time* ($p = .009$) and *invalid interactions* ($p < .001$). For the simple effects of the interaction effect for *SUS-Score*, the Dunn tests reveal significant differences for *occlusion* ($p = .029$), *pinching* ($p = .039$), and *tapping* ($p = .023$), but none for *no occlusion* ($p = .337$).

Regarding **H2**, we expect that hand occlusion yields higher usability for both interaction methods and conduct one-tailed t-tests and Wilcoxon tests between *occlusion* and *no occlusion* for both *pinching* (**H2.1**) and *tapping* (**H2.2**). Our analysis reveals a significant difference only between Tap+ and Tap- and only for *wrong confirms* ($p = .027$) and the *SUS-Score* ($p = .011$).

In **H3**, we expect *pinching* to have higher usability than *tapping*. Therefore, we group our results by the interaction method and perform one-tailed t-tests or Wilcoxon tests. The tests only reveal a significant difference between *pinching* and *tapping* for *menu time* ($p = .005$) and *invalid interactions* ($p < .001$).

Furthermore, we measured fatigue, perceived latency of the hand occlusion, and subjective rankings for efficiency and fun. Neither the Wilcoxon test for the Borg-scale yields a significant difference between *pinching* ($M = 1.5 \mid IRQ = 0.5 - 3.125$) and *tapping* ($M = 2{,}25 \mid IRQ = 1 - 3.125$) with $Z(29) = -0.105, p = 0.916$, nor for the latency questionnaire between the *pinching* method ($Mdn = 3 \mid IRQ = 2 - 5$) and the *tapping* method ($Mdn = 3.5 \mid IRQ = 2 - 4$) with $Z(29) = -1.356$, $p = 0.175$. The subjective rankings for efficiency for Pinch+ ($Mdn = 3 \mid IRQ = 2 - 4$), Pinch- ($Mdn = 2 \mid IRQ = 1 - 3$),

		Menu Time		Invalid Interactions		Wrong Confirms		SUS-Score	
Descriptive Statistics		*M*	*SD*	*M*	*SD*	*M*	*SD*	*Mdn*	*IQR*
Pinch+		15.57 *s*	3.59 *s*	.030	.049	0.40	0.81	78.750	59.375 – 85.000
Pinch-		14.50 *s*	3.34 *s*	.025	.033	0.37	0.85	81.250	66.250 – 88.125
Tap+		16.51 *s*	3.11 *s*	.186	.099	0.10	0.40	78.750	74.375 – 92.500
Tap-		16.80 *s*	5.31 *s*	.226	.131	0.47	1.17	75.000	62.500 – 92.500
Inferential Statistics		Statistic	*p*	Statistic	*p*	Statistic	*p*	Statistic	*p*
H1	ANOVA	$F(1,29)$		$F(1,27)$		$F(1,29)$		$F(1,29)$	
	Occ*Method	1.043	.316	2.070	.162	3.379	.076	**8.384**	**.007**
	Occlusion	0.563	.459	1.753	.197	2.500	.125	0.408	.528
	Method	**7.823**	**.009**	**122.897**	**< .001**	0.946	.339	0.540	.468
	T-Test	–		–		–		$t(29)$	
	Occlusion	–		–		–		**-2.303**	**.029**
	No Occlusion	–		–		–		0.897	.337
	Pinching	–		–		–		**-2.160**	**.039**
	Tapping	–		–		–		**2.406**	**.023**
H2	T-Test	$t(29)$		$t(27)$		$t(29)$		$Z(29)$	
2.1	Pinch+ Pinch-	1.637	.944	0.627	.732	0.297	.616	-1.944	.974
2.2	Tap+ Tap-	-0.289	.387	-1.424	.083	**-2.009**	**.027**	**-2.292**	**.011**
H3	T-Test	$t(29)$		$t(27)$		$t(29)$		$Z(29)$	
	Pinching Tapping	**-2.797**	**.005**	**-11.086**	**< .001**	0.972	.831	-0.365	.715

Table 1: Results of the descriptive and inferential statistical analysis of each hypothesis.
Pinch+ = Pinch with occlusion. Pinch- = Pinch without occlusion. Tap+ = Tapping with occlusion. Tap- = Tapping without occlusion.

Tap+ ($Mdn = 2.5 \mid IRQ = 1-4$), and Tap- ($Mdn = 2.5 \mid IRQ = 1-4$) show no significant differences by the Friedman test with $\chi^2(3) = 4.440, p = 0.218, N = 30$. For the subjective rankings for fun, the results for Pinch+ ($Mdn = 3 \mid IRQ = 2-4$), Pinch- ($Mdn = 2 \mid IRQ = 1-3$), Tap+ ($Mdn = 3 \mid IRQ = 1.75-4$), and Tap- ($Mdn = 2 \mid IRQ = 1-3.25$) also show no significant differences revealed by the Friedman test with $\chi^2(3) = 6.600, p = 0.086, N = 30$. The participants rated the task difficulty with $Mdn = 1 \mid IRQ = 1-1.25$.

7 Discussion

In **H1**, we expect an interaction effect between occlusion and the interaction method. Our analysis of *menu time*, *invalid interactions*, and *wrong confirms* revealed no such effect; however, we find a significant interaction effect in the *SUS-Score*. Thus, we **partially accept H1**. However, the exploratory investigation of the simple effect of the *SUS-Score* indicates a cross-over interaction against our expectations. As seen in Figure 3, occlusion seems to affect the *SUS-Score* for *pinching* negatively but positively for *tapping*.

We find the same effect in the analysis of **H2**, where we expect a positive effect of occlusion for *pinching* (H2.1) and *tapping* (H2.2). Similar to the first hypothesis, only the analysis of the *SUS-Score* yields significant results. The tests comparing Tap+ and Tap- indicated a positive effect of occlusion for *tapping*, leading to a partial acceptance of H2.2. However, the tests between Pinch+ and Pinch- did not show a positive effect for *pinching*, and the negative test statistic suggests a negative impact of occlusion for *pinching*. This is confirmed by the post hoc test of the simple effects of H1 and, thus, leads to a rejection of H2.1. As for an acceptance of H2, both H2.1 and H2.2 have to be accepted, we **reject H2**.

As these findings contradict our expectations of both the interaction effect between occlusion and the interaction method and the main effect of occlusion, we analyzed our data for unexpected side effects interfering with the effect of occlusion and the interaction method. Considering participants' comments from the post-questionnaire, we suspect the model-based occlusion approach caused interfering effects. Indeed, four participants explicitly stated that they found it cumbersome and irritating when part of the menu was occluded. This problem of unintentional occlusion, already acknowledged in previous studies [5, 1], affected *pinching* more adversely than *tapping*, according to our participants. This could have been a factor that led to the negative effect of occlusion on *pinching*.

An additional influential side effect could have been the level of acceptance for the occlusion. While 8 participants found the occlusion helpful for depth perception, 17 considered it irritating and unhelpful. This discrepancy suggests that not all participants perceived our model-based implementation as an "occlusion" but as a "shadow". Based on our observations and comments from the participants, we identified two technical issues that could have caused this low level of acceptance: Latency and the missing accommodative focus cue. The latency causes the model-based hand occlusion to lag behind the users' real hands, causing an offset of the occlusion while moving. The results of the latency question substantiate this issue. The lack of an accommodative focus cue, due to the fixed $2.0m$ focal distance of the Hololens 2 [38], resulted in a discrepancy in focus between real hands and the virtual model. This mismatch causes the occlusion to appear out of focus when the user is focusing on the user's real hands, which potentially causes irritations. Notably, participants did not undergo Hololens 2's eye-tracking calibration for interpupillary distance, and we did not screen for stereo blindness, strong eye dominance, or impaired visual acuity, limiting our understanding of these side effects. However, we do not find any indication that this effect had a different effect on *pinching* than on *tapping* and, thus, may have influenced the overall effect of occlusion but did not lead to the crossed interaction effect.

For **H3**, we expect higher usability for *pinching* compared to *tapping*. Our results indicate faster completion and fewer invalid interactions with *pinching*, supporting the hypothesis and leading to a **partial acceptance of H3**. However, this might be due to implementation issues with *tapping* rather than an inherently higher usability of *pinching*. On average, participants made 0.3 invalid *tapping* gestures for every valid *tapping* gesture and only 0.09 invalid *pinching* gestures for every valid *pinching* gesture. As some participants stated that the *tapping* was only registered when the gesture was performed

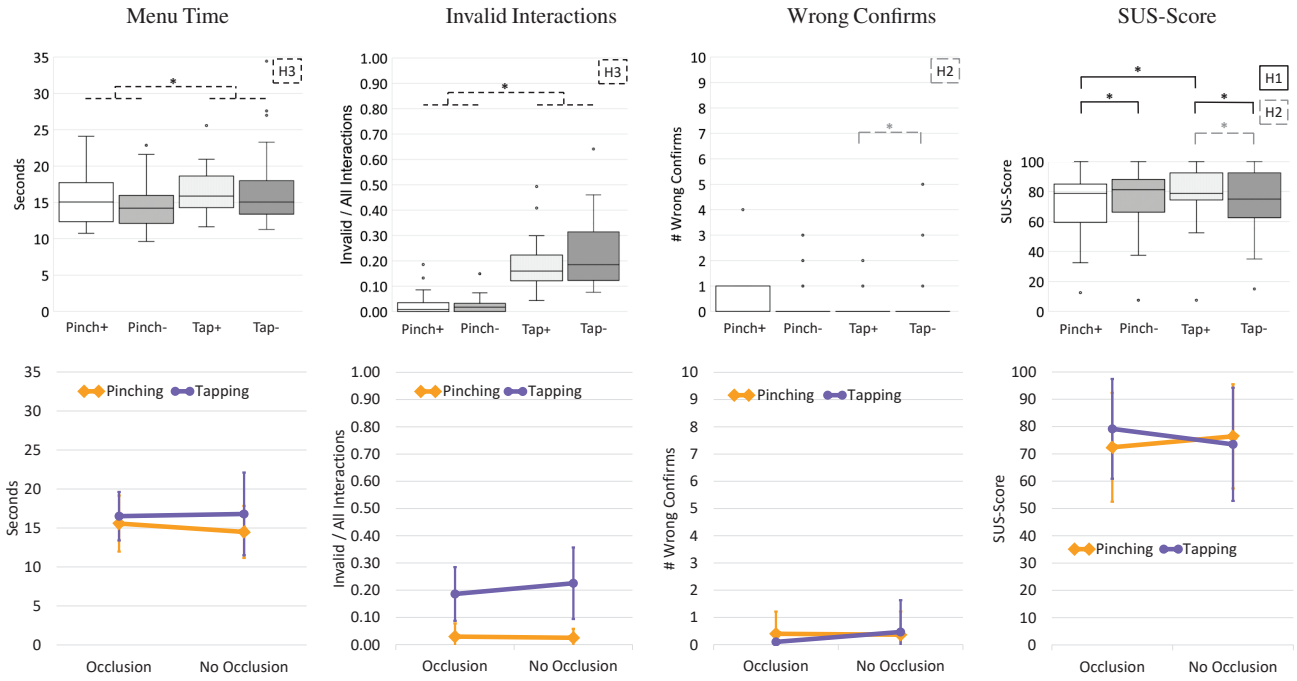

Figure 3: Descriptive analysis of four measures (top row) and the analysis of the interaction effect of the four measures (bottom row)

slowly, we expect this to be caused by the latency of the hand-tracking. This suggests that hand-tracking latency may have disproportionately impacted *tapping*, even if this is not captured by the latency questionnaire. However, the overall median of $Mdn = 3.5 \mid IRQ = 2 - 4.5$ indicates that the latency of our model-based hand occlusion was perceived negatively by the participants. Despite these findings, the analysis of the SUS score reveals no significant difference between the two methods. Considering *invalid interactions* as a technical error rate and *wrong confirms* as a task-related error rate, the non-significant t-test for *wrong confirms* implies that *pinching*'s advantage in *invalid interactions* might be technically driven. Further, the results of the Borg-scale indicate with $Mdn = 2.25 \mid IRQ = 0.84 - 3$ no major effect of arm fatigue. Regarding the significantly higher *menu time* for tapping, we assume that the higher count of overall interactions for *tapping*, due to the higher *invalid interactions*, also causes the higher *menu time*. To back this, we analyzed the videos and comments left by the participants and can not identify any other possible effect that may have had an impact on *menu time*. Thus, the higher *menu time* presumably results from the same technical issues as the *invalid interactions*.

The comments left by the participants indicate that the usability of these two methods comes down to personal preference. Both methods were positively annotated by adjectives like "intuitive", "easy", "fast", "precise", "fun", and "familiar", and negatively annotated like "imprecise", "tiring", and "laborious". This is also supported by the subjective rankings for efficiency and fun in the post-questionnaire. Neither the SUS Score, comments, nor rankings reveal a favorite interaction method. Therefore, we found indications for high usability for both interaction methods and that the differences stem from the issues of the *tapping* implementation.

8 Conclusion

In this work, we first implemented a model-based hand occlusion and then investigated the effect of this occlusion on usability in mid-air interaction by comparing two interaction methods. We found a cross-over interaction effect for the *SUS-Score* and a positive impact of occlusion on *tapping* on the *SUS-Score* and *menu time* but a negative impact on *pinching*. When examining our results closer, we found two possible variables, unintentional occlusion and acceptance of occlusion, which may have interfered with the actual effects and contributed to the analysis of how to account for these variables. Further, we found an effect for the interaction method: the *pinching* method has higher usability than the *tapping* method, which, however, seemed to be caused mainly by technical issues. Based on these observations, we discussed that there was no preference for either interaction method regarding usability.

9 Acknowledgements

This work is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant No.: 497135838.

References

[1] F. Argelaguet, L. Hoyet, M. Trico, and A. Lecuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *IEEVR '16: Proceedings of the IEEE Virtual Reality*, pp. 3–10, 2016. doi: 10.1109/VR.2016.7504682 5

[2] G. A. V. Borg. Psychophysical bases of perceived exertion. *Medicine & science in sports & exercise*, 14 5:81–337, 1982. doi: 10.1249/00005768-198205000-00012 4

[3] D. Bowman, E. Kruijff, J. J. LaViola Jr, and I. P. Poupyrev. *3D User interfaces: theory and practice*. Addison-Wesley, 2 ed., 2004. 2, 3

[4] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *I3D '97: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D graphics*, pp. 35–ff, 1997. doi: 10.1145/253284.253301 2

[5] P. Brandl, J. Leitner, T. Seifried, M. Haller, B. Doray, and P. To. Occlusion-aware menu design for digital tabletops. In *CHI EA '09: Proceedings of the ACM Extended Abstracts on Human Factors in Computing Systems*, pp. 3223–3228, 2009. doi: 10.1145/1520340.1520461 5

[6] J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189:4–7, 1996. doi: 10.1249/00005768-198205000-00012 3, 4

[7] G. Bruder, F. Steinicke, and W. Sturzlinger. To touch or not to touch? comparing 2d touch and 3d mid-air interaction on stereoscopic tabletop

surfaces. In *SUI '13: Proceedings of the ACM Symposium on Spatial user interaction*, pp. 9–16, 2013. doi: 10.1145/2491367.2491369 2

[8] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn. Fingartips: gesture based direct manipulation in augmented reality. In *GRAPHITE '04: Proceedings of the International Conference on Computer graphics and interactive techniques*, pp. 212–221, 2004. doi: 10.1145/988834.988871 2

[9] P. Chakravarthula, D. Dunn, K. Akşit, and H. Fuchs. Focusar: Auto-focus augmented reality eyeglasses for both real world and virtual imagery. *IEEE transactions on visualization and computer graphics*, 24:2906–2916, 2018. doi: 10.1109/TVCG.2018.2868532 1

[10] L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung. Touching the void: Direct-touch interaction for intangible displays. In *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2625–2634, 2010. doi: 10.1145/1753326.1753725 2

[11] J. Clark. *Designing for Touch*, chap. 4 - Gestures, pp. 129–182. A Book Apart, 1 ed., 2015. 1, 2

[12] S. Cooley, J. Mathew, and S. Paniagua. *Getting around HoloLens 2 - Grab using air tap and hold*, accessed 27.01.2023. `https://learn.microsoft.com/en-us/hololens/hololens2-basic-usage#grab-using-air-tap-and-hold`. 2

[13] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *SIGGRAPH '93: Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, p. 135–142, 1993. doi: 10.1145/166117.166134 1

[14] J. E. Cutting. How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers*, 29:27–36, 1997. doi: 10.3758/BF03200563 1

[15] DIN-33402-2. *Ergonomics - Human body dimensions - Part 2: Values*. Beuth Verlag, Berlin, 12-2020. 3

[16] C. Du, Y.-L. Chen, M. Ye, and L. Ren. Edge snapping-based depth enhancement for dynamic occlusion handling in augmented reality. In *ISMAR'16: Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 54–62, 2016. doi: 10.1109/ISMAR.2016.17 1

[17] K. Eveleigh, H. Ferrone, K. Semple, and CDiaz-MS. *Fingertip visualization*, accessed 25.01.2023. `https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/fingertip-visualization`. 2

[18] N. Feld and B. Weyers. Mixed reality in asymmetric collaborative environments: A research prototype for virtual city tours. In *WEVR '21: Proceedings of the IEEE Workshop on Everyday Virtual Reality*, pp. 250–256, 2021. doi: 10.1109/VRW52623.2021.00053 2

[19] Q. Feng, H. P. Shum, and S. Morishima. Resolving occlusion for 3d object manipulation with hands in mixed reality. In *VRST '18: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2, 2018. doi: 10.1145/3281505.3283390 1

[20] J. Fischer, B. Huhle, and A. Schilling. Using time-of-flight range data for occlusion handling in augmented reality. In *EGVE'07: Proceedings of the Eurographics conference on Virtual Environments*, pp. 109–116, 2007. doi: 10.2312/EGVE/IPT_EGVE2007/109-116 1

[21] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen. Extended pie menus for immersive virtual environments. *IEEE transactions on visualization and computer graphics*, 19:644–651, 2013. doi: 10.1109/TVCG.2013.31 1

[22] A. K. Hebborn, N. Höhner, and S. Müller. Occlusion matting: realistic occlusion handling for augmented reality applications. In *ISMAR'17: Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 62–71, 2017. doi: 10.1109/ISMAR.2017.23 1

[23] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1063–1072, 2014. doi: 10.1145/2556288.2557130 3

[24] I. P. Howard and B. J. Rogers. *Seeing in Depth Volume 2: Depth Perception*, vol. 2. Oxford University Press, 2002. 1

[25] H. Hua. Enabling focus cues in head-mounted displays. *IAO '17: Proceedings of the Imaging and Applied Optics*, 105:1–20, 2017. doi: 10.1109/JPROC.2017.2648796 1

[26] ISO 9241-11:2018. *Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. ISO/TC 159/SC 4, 03-2018. 3

[27] S. Jang, W. Stuerzlinger, S. Ambike, and K. Ramani. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, p. 3328–3339, 2017. doi: 10.1145/3025453.3025523 3

[28] M. Kim and J. Y. Lee. Touch and hand gesture-based interactions for directly manipulating 3d virtual objects in mobile augmented reality. *Multimedia Tools and Applications*, 75:16529–16550, 2016. doi: 10.1007/s11042-016-3355-9 1, 2

[29] F. Kistler and E. André. User-defined body gestures for an interactive storytelling scenario. In *INTERACT '13: Proceedings of the Human-Computer Interaction*, pp. 264–281, 2013. doi: 10.1007/978-3-642-40480-1_17 2

[30] L. Kreber and S. Diehl. A comparative evaluation of tabs and linked panels for program understanding in augmented reality. *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 29–38, 2023. doi: 10.1109/ISMAR59233.2023.00017 1

[31] L. Kreber, S. Diehl, and P. Weil. Idevelopar: A programming interface to enhance code understanding in augmented reality. *2022 Working Conference on Software Visualization (VISSOFT)*, pp. 87–95, 2022. doi: 10.1109/VISSOFT55257.2022.00017 1

[32] J. A. Leal-Meléndrez, L. Altamirano-Robles, and J. A. Gonzalez. Occlusion handling in video-based augmented reality using the kinect sensor for indoor registration. In *CIARP'13: Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 447–454, 2013. doi: 10.1007/978-3-642-41827-3_56 1

[33] N. Marquardt, K. Hinckley, and S. Greenberg. Cross-device interaction via micro-mobility and f-formations. In *UIST '12: Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 13–22, 2012. doi: 10.1145/2380116.2380121 2

[34] M. Maylyan, B. Holmes, R. Reynolds-Haertle, and C. Read. *HoloLens 2 gestures for authoring and navigating in Dynamics 365 Guides*, accessed 27.01.2023. `https://learn.microsoft.com/en-us/dynamics365/mixed-reality/guides/authoring-gestures-hl2`. 1, 2

[35] C. Meekhof, K. Sharkey, H. Ferrone, K. Eveleigh, D. Coulter, and S. Saltzman. *Direct manipulation with hands*, accessed 25.01.2023. `https://docs.microsoft.com/en-us/windows/mixed-reality/design/direct-manipulation`. 1, 2

[36] G. Norman. Likert scales, levels of measurement and the "laws" of statistics. *Advances in health sciences education : theory and practice*, 15:625–32, 2010. doi: 10.1007/s10459-010-9222-y 4

[37] K. C. Ong, H. C. Teh, and T. S. Tan. Resolving occlusion in image sequence made easy. *The Visual Computer*, 14:153–165, 1998. doi: 10.1007/s003710050131 1

[38] E. J. Paul, V. Tieto, S. Formicola, and D. Coulter. *Comfort*, accessed 25.01.2023. `https://docs.microsoft.com/en-us/windows/mixed-reality/design/comfort`. 5

[39] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *CHI EA '13: Proceedings of the ACM Extended Abstracts on Human Factors in Computing Systems*, pp. 955–960, 2013. doi: 10.1145/2468356.2468527 2

[40] D. Schmalstieg and T. Hollerer. *Augmented reality: principles and practice*, chap. 2.3 - Displays: Requirements and Characteristics, pp. 82–105. Addison-Wesley Professional, 1 ed., 2016. 1

[41] J. Schmidt, H. Niemann, and S. Vogt. Dense disparity maps in real-time with an application to augmented reality. In *WACV '02: Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 225–230, 2002. doi: 10.1109/ACV.2002.1182186 1

[42] X. Tang, X. Hu, C.-W. Fu, and D. Cohen-Or. Grabar: Occlusion-aware grabbing virtual objects in ar. In *UIST '20: Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 697–708, 2020. doi: 10.1145/3379337.3415835 1

[43] Y. Tian, T. Guan, and C. Wang. Real-time occlusion handling in augmented reality based on an object tracking approach. *Sensors*, 10:2885–2900, 2010. doi: 10.3390/s100402885 1

[44] A. Turner and V. Tieto. *Holographic Rendering*, accessed 25.01.2023. `https://docs.microsoft.com/en-us/windows/mixed-reality/develop/platform-capabilities-and-apis/rendering`. 2

[45] S. Uzor and P. O. Kristensson. An exploration of freehand crossing selection in head-mounted augmented reality. *ACM Transactions on Computer-Human Interaction*, 28, 2021. doi: 10.1145/3462546 2

[46] D. Zielasko, M. Krüger, B. Weyers, and T. Kuhlen. Passive haptic menus for desk-based and hmd-projected virtual reality. In *WEVR '19: Proceedings of the IEEE Workshop on Everyday Virtual Reality (WEVR)*, pp. 1–6, 2019. doi: 10.1109/WEVR.2019.8809589 2, 3, 4

[47] D. Zielasko, U. Skorzinski, T. W. Kuhlen, and B. Weyers. Seamless Hand-Based Remote and Close Range Interaction in Immersive Virtual Environments. In *GI Mensch und Computer*, 2018. doi: 10.18420/muc2018-ws07-0332 2