Increasing Realism of Displayed Vibrating AR Objects through Edge Blurring

MARCO KURZWEG, MAXIMILIAN LETTER, KATRIN WOLF, Berlin University of Applied Sciences and Technology, Germany

Many standard AR devices, such as the HoloLens 2, have limitations in displaying fast motions, like the ones required to visualize moving or vibrating objects. One reason for this is the low computing power compared to other technologies, resulting in frame rate drops. Further, established visualization enhancement methods, such as anti-aliasing, cannot be applied because of their high computational demands. Therefore, we have looked at possible alternatives on the HoloLens 2 for displaying vibrations more realistically as long as these technical limitations exist. We have chosen to examine vibrations as they are widely used for different use cases, like creating feedback, communicating the success of interactions, and generating a better scene understanding.

In a user study, three different effects were evaluated against a baseline method, which was the representation of a vibration using a sinus function to calculate the displacement of the object. We found that an effect where the edges of the AR object are blurred (continuously with changing intensity) is perceived as significantly more realistic than other effects and the baseline method.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; **Empirical studies in HCI**; **Interaction design**.

Additional Key Words and Phrases: visual effect, vibration

ACM Reference Format:
Marco Kurzweg, Maximilian Letter, Katrin Wolf. 2023. Increasing Realism of Displayed Vibrating AR Objects through Edge Blurring. In *Mensch und Computer 2023 (MuC '23), September 3–6, 2023, Rapperswil, Switzerland*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3603555.3603570

1 INTRODUCTION

In recent decades, technological improvements in hardware and software have led to ever-increasing fidelity in visual representation [19]. While some technologies can take advantage of these advances, portable standalone technologies, such as optical see-through Augmented Reality (AR) glasses, are not able to reach the same level of quality that stationary systems can achieve. In this work, we focus on that device category, which is currently limited by available hardware.

For example, the Microsoft Hololens 2, one of the dominant AR headsets on the market, has a battery of limited capacity to ensure a usable form factor. This, along with the integrated processor, existing display technology, and protection from potential heat exposure, results in limited performance. As a result, some visual effects that are well-established on other platforms are difficult to implement. For example, Microsoft states that anti-aliasing, a post-processing method to smooth motion, does not run on the Hololens 2 and should not be attempted, as the computational requirements can lead to large performance drops [18]. Instead, effects could be imitated using shaders [18]. While the example of anti-aliasing is a computationally intensive process, the same problem arises with processes that are typically not computationally demanding, such as visually representing a vibrating virtual object. A visual vibration refers to a non-physical vibration that visualizes the expected characteristics of one, thereby inducing the feeling of experiencing a vibration. In this work, we aim to increase the realism of moving, in particular vibrating, objects for the state-of-the-art AR headset Hololens 2.

Besides performance limitations, the inclusion of the real world in AR results in some alternative approaches to display motions and vibrations not being transferable to this technology. For example, in fully virtual environments, the scene camera could be displaced to emulate vibrations, driving on uneven ground, walking, or shaking. When taking a look at AR, however, this technique does hardly work as the real world visible to the user cannot be altered. Therefore, such an effect can only be accomplished by manipulating the virtual objects in the scene directly.

The most common way of manipulating a virtual object to represent a vibration is to use a sinus function for movement. This was used, for example, to suggest the haptic sensation of stiffness [15]. In this approach, which was also carried out on a HoloLens 2, the virtual object vibrated with a very low frequency. This is due to the fact that technical limitations have already become apparent here and limit this effect. At the same time, it shows that certain visual effects and the resulting visual feedback can be used to generate haptic sensations, which is a standard method in research (i.e., weight [27], shape [3], size [4], or stiffness [42]). Further, visual vibrations can also be used to emphasize some events, such as the chiming of a cartoon clock (known as Non-Photorealistic rendering) [11], or to give the impression of walking like a heavy creature by shaking the scenery [38]. Besides, visual vibrations can be a form of feedback to the user when they touch a virtual object [13] or are used to create or strengthen pseudo haptic feedback [12]. As vibration is a standard haptic feedback technique in human-computer interaction, which cannot be supported by AR glasses that support audio and vision only, the visual representation of vibration feedback gains importance here. As described earlier, AR glasses lack quality when it comes to displaying vibrations, which motivates the research presented here.

In a technical evaluation with the HoloLens 2, we found that the effective frame rate, and thus the frequency at which vibration can be presented, is limited to roughly 40 Hz, which we discuss in more detail in Section 3. Due to this and the lack of other common approaches (i.e., anti-aliasing), this work investigates which visual effects could improve the visual representation of a vibration. While making AR devices more powerful would also be possible, this could lead to a decrease in runtime and increased costs. Visual effects are, therefore, an alternative. At the same time, a solution working on low-power devices is more universal and scalable.

To determine which visual effects are most likely to be used to represent vibrations, we have used various effects. These are known from other fields or everyday experiences. These were then tested in a controlled lab study against a conventional method of representing a vibration by moving the vibrating object at a high frequency and against a control condition in which no motion was present. In our study, we found that an effect applied on the virtual object by being blurred at the edges with a shader performed statistically better than our other tested effects, the baseline, and the control condition.

2 RELATED WORK

In this work, we explore which effects can be used to improve the visual representation of a vibrating virtual object. Therefore, we reviewed methods to improve the visual representation and performance of virtual objects. These were technical solutions as well as approaches based on users' perceptions.

2.1 Technical Challenges

Presenting virtual environments and experiences on screen has benefited from increasing the resolution and pixel density of displays over the last years [19]. This is true for all kinds of display-using media, including immersive technologies like VR and AR. In order to achieve convincing image quality while using a reasonable amount of computing power, several methods are used to improve visual perception as much as possible without losing too much quality [19]. Most of the investigated methods aim at optimizing rendering performance when limited time and resources are available [2, 43]. Depending on the target platform, methods can be applied more or less easily to that platform. A challenging platform for using various effects is immersive AR, which requires two images of a scene in high as possible resolution, a high frequency of frames, and aesthetics merging between virtual and real environments. Because of these properties, AR is especially dependent on looking at established methods for graphic optimization, such as Foveated Rendering, Ray-Tracing, Level of Detail, and Post-Processing effects.

2.2 Rendering Techniques

Foveated Rendering makes use of different aspects of the fovea by only rendering the region where our eyes focus on in full resolution and displaying everything else in lower density [40]. Kim et al. focussed on designing a dynamic foveated AR display examining the fusion of real and virtual content, which will influence the rendering results [17]. As they found out, controlling the degree of fusion for having images of different qualities in different regions of vision is still challenging.

Ray-tracing is an image synthesis method where a 2D image is created from a 3D world [6, 10, 33]. While it creates realistic lighting, occlusion, and shadows in a scene, it is rather computing intensive [29]. Therefore, it was not suitable to be used in interactive real-time applications, and instead, other graphic pipelines were used. With modern Graphics Processing Units (GPUs), it became possible to enable ray-tracing for a specific part of the rendering process, such as reflections [1, 28]. As these are usually unavailable for AR devices, research was conducted to explore optimizations on ray tracing to make it feasible for AR applications [29].

Another example is Level of Detail (LOD), which alters the quality (complexity of models, resolution of textures) of objects that have to be rendered based on the distance they are in the scene [14]. LOD has been explored for AR, exemplary for glanceable AR interfaces or 3D games and other smartphone applications [8, 37]. Daskalogrigorakis et al. investigated using LOD for AR interfaces, as complex interfaces often result in unintentional eye gaze interaction and selection [8]. To solve this, they used glanceable 2D interfaces with compact information and the possibility to activate additional interactions like touch input. Syaputra et al. explored the limitation of increased processor workload and increased devices' heat when running AR applications on smartphones [37]. They used the distance between the camera and marker to change the LOD of presented virtual objects. They were able to reduce the processor's workload and reduce the temperature of the device.

2.3 Post-Processing Methods

Other than the named examples, post-processing effects are methods applied at the end of the rendering pipeline, modifying the so far created images. Anti-Aliasing is one of those post-processing effects that visually smooth edges in an image by recoloring adjacent pixels. While such effects are commonly used on laptops and desktop computers, their use is usually demanding in computer power and therefore discouraged to use on AR devices [18, 29].

While all these methods provide good rendering results on other platforms, they are currently not fully implementable for AR. Besides these approaches, all relying on technical measures, there exist effects based on the perceptions and experiences of users. For example, Motion Parallax induces the illusion of depth by translating two-dimensional background layers at different speeds, moving content slower the further it is away [9, 26, 31].

Similarly, non-photorealistic rendering uses exaggerated presentations of actions in a scene to underline what is happening. Inspired by comics and cartoons, Kawagishi et al. developed a deformation algorithm that emphasizes movements in a scene [16]. They propose different techniques to support the movement: for example, adding lines along the path the object takes, leaving after-images of the moving object behind, and deforming the object's contour at its rear side. Every effect was further experimented with by using different properties, i.e., amount, length, and color of the added lines. Similar effects have been successfully used in racing games, adding blur to the background or lines at the rim of the vehicle to emphasize its speed [25, 32, 39].

Another blurring effect closely related to the cartoon blur is the motion blur. This effect is also commonly used for the perception of objects in motion. Motion blur is the result of combining relative motion and light integration taking place, for example, in films [20]. Navarro et al. presented different methods to simulate motion blur and explained how these methods work. As motion blur can require high computational costs, Park et al. investigated how motion blur can be used for rendering in AR. They found warping of images under 3D perspective as an alternative to 3D rendering for creating blurred images at a very low computational cost [22].

As such approaches – of using visual effects to induce motion or vibration perception – are usually not computationally demanding, they might be a starting point to investigate how to represent motions and vibrations otherwise – due to current technical limitations – not possible in the context of AR. We address this research gap in the presented paper.

3 TECHNICAL REQUIREMENT IDENTIFICATION & EFFECT DESIGN

Visual effects were designed with the intent to reach better realism of motion/vibration visualization on AR state-of-the-art devices, such as the Hololens 2. As that design bases, inter alia, on properties and limitations of the device, hardware limitations of the Hololens 2 will be discussed, as well as how they were identified and how they differ from other devices and applications. Lastly, the motivation for the chosen effects and the choice of a baseline comparison condition is presented.

3.1 Technical Limitations

For the technical evaluation of the Hololens 2, we first examined the frame rate of the device. This was done by recording time stamps in a CSV file during the visual vibration of a virtual object. The visuals were created by moving the virtual object along one axis with a sinus function. The recorded time stamps were then used to calculate the effective frame rate of the device. This evaluation showed that the frame rate of the HoloLens 2, in our application, was roughly 40 Hz most of the time (78 %). This is in line with a movement frequency of 40 Hz, resulting in the object being static and its movement not being noticeable. As a result, the movement of the virtual object, even at small amplitudes (2 mm), looked more like flickering since the steps in position changes were too large to represent smooth motion.

There exist a variety of effects that could potentially improve the visuals, for example, motion blur and anti-aliasing. While techniques requiring a lot of computational power (anti-aliasing) are not supported on HoloLens 2 and are therefore not recommended by Microsoft [18], other alternatives like shaders are advised and used for some effects presented in this work.

Concerning these existing hardware limitations, we examine which alternative approaches can be used to create a visual vibration and how to achieve the most realistic representation.

(a) (b) (c) (d)

Fig. 1. Images of the baseline condition and the three effects used within the study: (a) shows the baseline condition, where the movement of the gray cuboid was calculated using a sinus function. (b) is an extension of the baseline condition, using three instead of a single virtual cuboid, each shifted in its phase of the sinus function movement. (c) displays the effect of blurring the edges of the virtual object itself instead of the background. The shader was applied with a continuously changing intensity from 0.1 cm to 0.6 cm. (d) presents the visual effect of the background blur where a shader was used to blur the checkerboard texture where the cuboid was placed on. Equal to (c), the intensity of the shader continuously changed.

3.2 Visual Effects Design

In the following, we describe the baseline method of creating a visual vibration of an object, as well as the different visual effects proposed as alternatives. Further, we describe how the effects were chosen, designed, and implemented. For every condition, the participants looked at a virtual cuboid placed on a plane with a checkerboard texture; see Figure 1.

3.2.1 Sinus Motion. As our baseline condition to which we will compare all used visual effects, we have chosen to present the vibration of the virtual object by a displacement animation defined through a sinus function with an amplitude value of 0.6 cm; see Figure 1a. We conducted the pre-study with a team of 6 researchers. Different amplitudes (0.2 cm, 0.4 cm, 0.6 cm, 0.8 cm, 1 cm) were tested, and 0.6 cm was found to be the best recognizable and, at the same time, not too much to be perceived as unrealistic. The linear increase of 0.2 cm between the amplitudes was chosen based on Weber-Fechner law and Steven's power law [21, 34], and 0.2 cm was found to be a small step but still being recognizable by the participants. The frequency chosen for the sinus function was 40 Hz due to the limitation of the device, as mentioned before. The following formula was used for the calculation of the displacement:

$$x = x_0 + \sin(time \cdot frequency \cdot (2 \cdot \pi)) \cdot amplitude$$

This method is commonly known as the representation method for calculating vibrations [11–13, 38]. It is used in previous research, including visual vibration representation on a HoloLens 2 [15], as well as in this and other scientific fields to describe a vibration (i.e., as a visual indicator to provide feedback [13, 38]). Because of these varied and frequent uses, we took this established method as a baseline for comparison. This method was also used while calculating the effective HoloLens 2 frame rate.

3.2.2 Multiple Objects. For the condition of multiple objects, we stacked three objects with the sinus baseline effect on top of each other and started their animations consecutively to strengthen the perceived vibration in comparison to the baseline; see Figure 1b. These two additional objects were copies of the base virtual cuboid. The calculation of the movement was the same as for the baseline, as well as the frequency (40 Hz) and the amplitude (0.6 cm). Still, every additional object was presented with a periodical phase shift of half, respectively, a third of the phase compared to the main object. This means that the displacement starts at the same time for every single cuboid, but due to the phase shift, they are all at another state of the sinus function and, therefore, at another place:

$$x{=}x_0 + \sin((time - \Delta time/2) \cdot frequency \cdot (2 \cdot \pi)) \cdot amplitude$$
$$x{=}x_0 + \sin((time - \Delta time/3) \cdot frequency \cdot (2 \cdot \pi)) \cdot amplitude$$

With this effect, it still looked like there was only one object, but the movement did not look that erratic anymore. With this smoothing of the movement, we aimed to mimic the effect of anti-aliasing on a lower level, but also without the need for high computational resources; see Section 2. While anti-aliasing smooths edges by recoloring pixels, we smoothed the movement by adding two objects, which had the effect that more motion states were shown on the screen. Therefore, the jumps between positions of one square were reduced using three squares, resulting in a more continuously looking motion.

3.2.3 Object Blur. One effect we used for representing a vibrating object is blurring the virtual object's edges in changing width, while the object itself did not move; see Figure 1c. Therefore, we took a shader where a float value was used for the displacement distance (blurring) of the object's edges. The strength of the blur shader could be adjusted to blur the edges of the virtual object over a range (from 0.1 to 0.6 cm). The maximum value here (0.6 cm) was the same as the amplitude used for the baseline. The frequency of displaying the different strengths was set to the frame rate of the device, resulting in 40 Hz. We used the object blur as this effect is inspired by real-life experiences, such as the vibration of strings of a musical instrument, which move so fast that we only perceive them as unsharp or blurred. We also know motion blur from photography and video frames when capturing a fast-moving object, which has been used in previous work as non-photorealistic motion blur [16] or motion blur [22]; see Section 2. Even if motions blur is commonly used for continuously fast-moving objects, we found it valuable to investigate if this also works for a fast motion in place.

3.2.4 Background Blur. For this effect, the virtual cuboid did not move at all. Instead, we blurred the plane the virtual object was placed on, with changing strengths using a shader; see Figure 1d. Within the shader, a float value was used for the displacement distance (blurring) of the texture to the side. This value set the strength of the blur effect and ranged from 0.1 to 0.6 cm. The maximum value of this range (0.6 cm) was taken from the amplitude of the sinus movement, as also done for the object blur. The strength of the blur effect was set in the update function and changed related to the frequency of the device, which in this case was 40 Hz. By blurring the plane, we aimed to create the effect known from the vection illusion [30] or also known from motion parallax [9, 31]; see Section 2. While the mentioned methods have the effect that an object is perceived in motion even if standing still, we aimed at investigating if a known illusion, like vection, also can be applied to virtual objects.

For all effects, we expected that the sensation of visually perceiving a vibrating object would occur by providing well-known vibration-representing visual effects. Besides the standard method of Sinus Motion, we chose the three alternative effects, as they are all related to motion in other closely related contexts. In addition, they can all be implemented with shaders or techniques that do not rely on high computational power.

4 USER STUDY

We aimed to explore how the visual effects simulating an AR object vibration can be perceived as more realistic than the baseline implementation that suffers from technical limitations. Therefore, we determined the following research question:

- How can we improve the impression of visually perceiving a vibration of an object displayed on AR glasses without anti-aliasing and a frame rate limitation of 40 Hz?

4.1 Experiment Design

As our goal was to explore if we can increase the realism of visually perceived vibrations and determine through which effect this is possible, we tested different known visual effects within our study. We designed a controlled experiment with a 5x1 within-subjects design and the independent variables VISUALEFFECTS (*sinusMotion*, *multipleObjects*, *objectBlur*, *backgroundBlur*, *no effect*). The VISUALEFFECTS align with the effects presented in Section 3. Additionally, *no effect* was added as a control condition where no effect was displayed.

The dependent variables were, VIBRATIONRECOGNITION (to explore if the effects worked as intended), PERCEIVEDSTRENGTH (to measure how strong the vibration appeared to the participants), REALISM (as a measure if the vibration was perceived as realistic), and REALISTIC APPEARANCE FACTORS (to identify reasons that supported or prevented a realistic look of the vibration).

4.2 Measurements

To explore if the effects work in the intended way, we first asked if it looked like the gray cuboid was vibrating with a binary choice (Yes/No). To answer our research question, we asked participants about the PERCEIVEDSTRENGTH and the REALISM of the vibrations they saw. Therefore, we used two single-item questions, like done in prior work [23, 24, 35]. This resulted in the two following questions:

- As how strong did you perceive the gray cuboid's vibration?
- How real did the vibration of the gray cuboid look?

Both answers had to be answered on a 7-item Likert scale. For PERCEIVEDSTRENGTH, the Likert scale ranged from *no movement at all (1)* to *very strong (7)*, and for REALISM, from *not real at all (1)* to *absolute real (7)*.

Further, we asked about reasons that supported or prevented a realistic look of the vibration in a semi-structured interview:

- What supported the realistic look of the movement?
- What limited the realistic look of the movement?

4.3 Participants

We recruited 20 participants (6 female, 13 male, and 1 other) with an age range from 20 to 39 years and an average of 26,15 years (SD = 4,61). The participants were recruited via mailing lists and had professional backgrounds, such as computer science or robotics.

4.4 Apparatus

Our system utilized the HoloLens 2 as the main device. The participants could see a gray cuboid of 10 x 10 cm in size and a height of 1 cm within the active scene. The gray cuboid was placed directly on a flat virtual surface with a size

Fig. 2. Left: Participants sat at an empty table during the study. This illustration shows where the virtual content was located on the table, which the participants could see through the HoloLens 2. Right: The view the participants had while looking through the HoloLens 2 before they started a condition. Therefore, the virtual blue button to start a condition is present.

of 30 x 30 cm and a checkerboard texture. On the right side of the cuboid, a virtual button was placed to go through different conditions; see Figure 2. The conditions were the visualEffects (*sinusMotion*, *multipleObjects*, *objectBlur*, *backgroundBlur*, and *no effect*) used in this work, and their design and implementation can be seen in Section 3. Their order was set up in the application. After finishing one state, the application automatically switched to the following condition. The button which started a new condition was hidden while a visualization was being displayed. This further focused on the cuboid and its visual vibration, as distractions were reduced during the effect. A notification sound was played through the speakers of the Hololens 2 at the end of each condition, signaling to the participant that the condition was finished. The questionnaires were filled in on a separate computer.

4.5 Task & Procedure

The experiment was conducted as a lab study. First, the participants were introduced to the study's purpose and then asked to agree to a consent form. After filling in a demographic questionnaire, participants started with the study. In each condition, the participants pressed the start button when ready and looked at the scene with the gray cuboid on the checkerboard-looking surface. The distance to the vibrating object was defined by the distance of the AR glasses to the surface on which the virtual object was placed. The individual participants were always sitting in the same position during the study, so the distance was always similar for a single participant for all conditions. Between each participant, the distance could slightly differ, depending on their body sizes. The order of the different conditions was counter-balanced using a Latin square design [5]. After each condition, participants filled in the questionnaire and answered the semi-structured question. Participation in the study took approximately 15 minutes.

5 RESULTS

We first analyzed our quantitative data to identify if movements of the gray cuboid were perceived at all for the different visualEffect conditions (*sinusMotion*, *multipleObjects*, *objectBlur*, *backgroundBlur*, and *no effect*).

Further, we investigated if the visualEffect conditions affected the perceivedStrength and the realism of the feedback. Subsequently, we analyzed the qualitative data to understand our quantitative results better.

5.1 Quantitative Results

For analyzing the quantitative results of the vibrationRecognition, we used a chi-squared test. For the vibrationStrength as well as for the vibrationRealism, we used a Friedman test to identify significant differences.

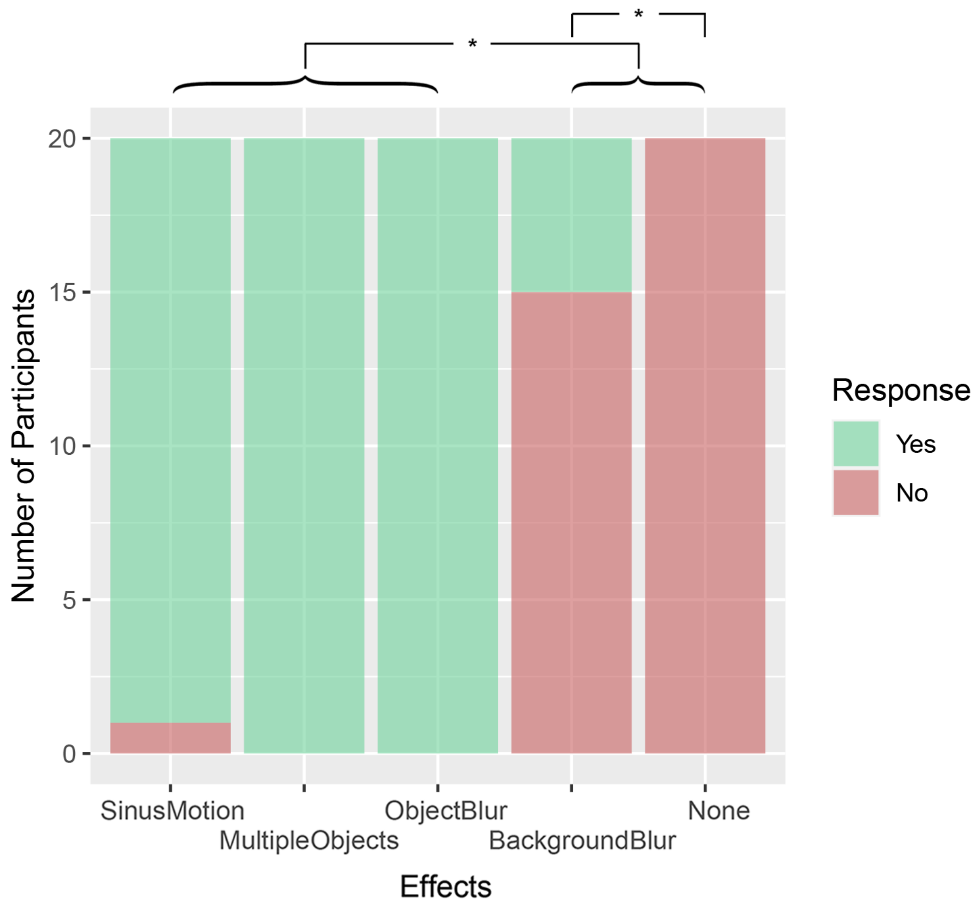

Fig. 3. Participants' answers whether the gray cube was perceived as moving. The y-axis shows the number of participants, and the x-axis shows the different visual effects used.

Post-hoc analysis with Wilcoxon Signed-Rank tests was conducted with a Bonferroni correction for the p-value, resulting in a significance value of 0.01.

5.1.1 Vibration Recognition. First, it was checked if the different effects worked as intended and a vibration was perceived.

Regarding the two VISUALEFFECTS *objectBlur* and *multipleObjects*, the virtual gray cuboid was found to be vibrating in 100% of all cases. Further, the *sinusMotion* was perceived as vibrating in 95% and the *backgroundBlur* in 25% of all cases. For the condition *no effect*, the cuboid was never perceived as a vibrating object; see Figure 3. A chi-squared test indicates significantly better vibration recognition for *sinusMotion*, *objectBlur*, and *multipleObjects* compared to *backgroundBlur* and *no effect* and for the *backgroundBlur* compared to *no effect*; see Figure 3 and Table 1.

Sample 1 (S1)	Sample 2 (S2)	*Mdn* (S1)	*Mdn* (S2)	W (effect-size)	p-value
SinusMovement	MultipleObjects	1.0	1.0	0.1601	1.0
SinusMovement	ObjectBlur	1.0	1.0	0.1601	1.0
SinusMovement	**BackgroundBlur**	**1.0**	**0.0**	**0.7144**	**< .001**
SinusMovement	**No effect**	**1.0**	**0.0**	**1**	**< .001**
MultipleObjects	ObjectBlur	1.0	1.0	0.1203	1.0
MultipleObjects	**BackgroundBlur**	**1.0**	**0.0**	**0.7746**	**< .001**
MultipleObjects	**No effect**	**1.0**	**0.0**	**1.0**	**< .001**
ObjectBlur	**BackgroundBlur**	**1.0**	**0.0**	**0.7746**	**< .001**
ObjectBlur	**No effect**	**1.0**	**0.0**	**1.0**	**< .001**
BackgroundBlur	**No effect**	**0.0**	**0.0**	**0.5673**	**< .001**

Table 1. Results for the vibrationReconition of chi-square tests. Sample 1 and 2 are possible combinations of visualEffects. All possible combinations are checked against each other to see if they have significant differences. Significant results are displayed in bold font.

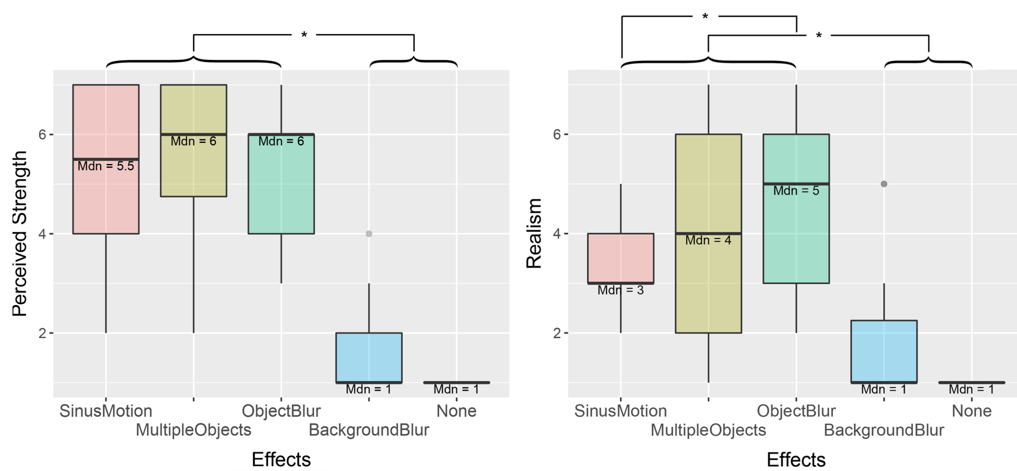

Fig. 4. Boxplots showing the participants' answers of the perceived strength (left) and the perceived realism (right) for each visual effect. The y-axes show the values of the answers, ranging from 1 (low) to 7 (high). The x-axes show the visual effects.

5.1.2 Vibration Strength. After the vibration recognition analysis, we gained an overview of the dependent variable perceivedStrength. A Friedmann test indicated a significant difference for the different visualEffects ($\chi^2(4) = 61.264$, $p < .001$).

Post-hoc analysis performed with Wilcoxon Signed-Rank tests indicated a statistically significant higher perceived-Strength for the visualEffects *sinusMotion*, *multipleObjects*, and *objectBlur* in comparison to *backgroundBlur* and *no effect* (always with $p < .001$), see Figure 4 and Table 2.

5.1.3 Vibration Realism. For the dependent variable realism, a Friedman test indicated a significant difference for the independent variable visualEffect ($\chi^2(4) = 55.880$, $p < .001$).

Post-hoc analysis performed with Wilcoxon Signed-Rank tests indicated a statistically significant higher realism for the visualEffects *sinusMotion*, *multipleObjects*, and *objectBlur* compared to *backgroundBlur* and *no effect* (always

Sample 1 (S1)	**Sample 2 (S2)**	*Mdn* **(S1)**	*Mdn* **(S2)**	**r (effect-size)**	**p-value**
SinusMovement	MultipleObjects	5.5	6.0	-0.055	0.802
SinusMovement	ObjectBlur	5.5	6.0	-0.046	0.834
SinusMovement	**BackgroundBlur**	**5.5**	**1.0**	**-0.840**	**< .001**
SinusMovement	**No effect**	**5.5**	**1.0**	**-0.877**	**< .001**
MultipleObjects	ObjectBlur	6.0	6.0	-0.031	0.887
MultipleObjects	**BackgroundBlur**	**6.0**	**1.0**	**-0.865**	**< .001**
MultipleObjects	**No effect**	**6.0**	**1.0**	**-0.877**	**< .001**
ObjectBlur	**BackgroundBlur**	**6.0**	**1.0**	**-0.876**	**< .001**
ObjectBlur	**No effect**	**6.0**	**1.0**	**-0.879**	**< .001**
BackgroundBlur	No effect	1.0	1.0	-0.573	0.011

Table 2. Results for the perceivedStrenth of Wilcoxon Signed Rank tests (a Bonferroni correction for the p-value of 0.01). Sample 1 and 2 are possible combinations of visualEffects. All possible combinations are checked against each other to see if they have significant differences. Significant results are displayed in bold font.

Sample 1 (S1)	**Sample 2 (S2)**	*Mdn* **(S1)**	*Mdn* **(S2)**	**r (effect-size)**	**p-value**
SinusMovement	MultipleObjects	3.0	4.0	-0.305	0.171
SinusMovement	**ObjectBlur**	**3.0**	**5.0**	**-0.579**	**0.009**
SinusMovement	**BackgroundBlur**	**3.0**	**1.0**	**-0.786**	**< .001**
SinusMovement	**No effect**	**3.0**	**1.0**	**-0.883**	**< .001**
MultipleObjects	ObjectBlur	4.0	5.0	-0.328	0.142
MultipleObjects	**BackgroundBlur**	**4.0**	**1.0**	**-0.772**	**< .001**
MultipleObjects	**No effect**	**4.0**	**1.0**	**-0.854**	**< .001**
ObjectBlur	**BackgroundBlur**	**5.0**	**1.0**	**-0.850**	**< .001**
ObjectBlur	**No effect**	**5.0**	**1.0**	**-0.877**	**< .001**
BackgroundBlur	No effect	1.0	1.0	-0.515	0.021

Table 3. Results for the realism of Wilcoxon Signed Rank tests (a Bonferroni correction for the p-value of 0.01). Sample 1 and 2 are possible combinations of visualEffects. All possible combinations are checked against each other to see if they have significant differences. Significant results are displayed in bold font.

with $p < .001$). In addition, the visualEffect *objectBlur* was rated with a significantly higher realsim compared to *sinusMotion*, see Figure 4 and Table 3.

5.2 Qualitative Results

The qualitative data was coded using Grounded Theory [36]. Axial and selective coding was applied, building categories according to the questions that asked for positive or negative perceived aspects of a system [36]. Two researchers did the coding independently of each other and discussed their results afterward to develop common codes. The goal was to gain possible explanations for why one effect outperformed another. The qualitative results are separated into factors that supported a realistic look of the vibration and factors which prevented a realistic look of the vibration, which were pre-structured by the semi-structured interview questions. In addition, we will distinguish between the different effects to better understand possible differences between them.

Since answering the question was optional, not every participant always answered it. Hence, we had a total of 106 responses. 20 of these 106 responses will not be taken into account from now on, as they refer to the VISUALEFFECT *no effect*, where no motion was perceived, which can be found in Figure 3. Therefore, we consider only the remaining 86 responses. 43 of these 86 responses addressed both realism-supporting and realism-limiting factors.

5.2.1 Realism Supporting Factors. Two arguments could be identified as supporting factors for a realistic appearance of vibrations. These two points were the object's movement and the cuboid's difference from the static background.

The object's movement was mentioned 40 out of 43 times as a possibility to support a realistic vibration appearance. This was mainly the case for *multipleObjects* (10 out of 40 cases), *objectBlur* (14 out of 40 cases), and *sinusMotion* (12 out of 40 cases). In addition, the object's movement was named four times as a supporting factor for *backgroundBlur*. Exemplary answers:

- the small but fast movement (P.16, *sinusMotion*)
- The movement looked a bit smoother (P.17, *multipleObjects*)
- Mostly like I imagine a vibration, not a movement to the side, but a movement in place (P.19, *objectBlur*)
- just small movements like in real life (P.15, *backgroundBlur*)

The remaining 3 of the total 43 responses on supporting factors were related to the contrast of the gray cuboid from the static background. This point was once mentioned for *sinusMotion* and 2 times for *multipleObjects* :

- The area the gray cuboid was placed on was not moving (P.2, *sinusMotion*)
- As the background was not moving, the motion of the gray object was perceived better (P.18, *MultipleObjects*)

5.2.2 Realism Limitating Factors. For the VISUALEFFECT *sinusMotion*, 8 responses regarding limiting factors were given by the participants. In 4 out of these eight answers, it was mentioned that the seen vibration was perceived as too strong, and in the other 4 cases, it was mentioned that the motion looked more like a flickering than a vibration:

- The movement was too strong (P.4, *sinusMotion*)
- It looked more like a flickering object than a vibrating one (P.6, *sinuMotion*)

Three different factors were named as limiting factors for the VISUALEFFECT *multipleObjects* in 7 out of the 43 responses. These were that the movement of the vibration was too strong (4 times) and the movement of the vibration was too weak (1 time). In addition, it was mentioned that the movement was not convincing and looked more like flickering:

- Too less movement (P.17, *multipleObjects*)
- The motion was too strong. Seemed very cartoonish (P.20, *multipleObjects*)
- The motion seemed very choppy. Maybe the framerate was too low (P.8, *multipleObjects*)

Also, three different limiting factors were named for the *objectBlur* with 8 out of the 43 responses. Just as with the sinus motion, the object's movement here was perceived as both too strong (3 times) and too weak (2 times). Further, it was mentioned that the motion looked too static (3 times):

- The movement was too weak and too uniform (P.13, *objectBlur*)
- The motion was too strong (P.4, *objectBlur*)
- For the strength of the vibration, it looked too static (P.10, *objectBlur*)

10 out of 43 responses were given for the VISUALEFFECT *backroundBlur*. All these responses related to the point that the effect was recognized, resulting in perceiving a motion of the background but not in perceiving the cuboid vibrating:

- Just the background was in motion and not the object (P.8, *backgroundBlur*)

Further, across all conditions, it was mentioned several times (in 10 out of 43 cases) that a missing sound limited the experience of perceiving a vibrating object:

- No sound for the friction with the surface (P.12, *objectBlur*)
- A sound was completely missing (P.11, *multipleObjects*)

6 DISCUSSION

In this work, we explored how the impression of visually perceiving a vibration of an object displayed on state-of-the-art AR glasses could be improved. Therefore, we investigated the perceived realism of three VISUALEFFECTS known from other contexts, compared to a standard method of visualizing a vibration, namely moving the object back and forth via a sinus function. The same function was used for the representation of the vibration by three objects, but the individual objects were shifted in their phase. For the effects where the object itself or the background was blurred, the object did not move, but through the animated blur effects, a movement was intended to be simulated. Instead, the blurring effect was displayed in continuously changing intensity to simulate a vibration. In addition, all effects (*sinusMotion*, *multipleObjects*, *objectBlur*, *backgroundBlur*) were also tested against *no effect* during the study. However, since *no effect* only served as a control condition, the results of this effect are not considered in the discussion. In terms of perceived realism, the object with animated blur outperformed the standard method, as confirmed by the participants' qualitative feedback. Both quantitative and qualitative results are discussed here. Based on this, considerations are given on which effects might be useful to create visually represented vibrations. We lastly regard the limitations of our design and implementation and how these might be addressed by future research or improved hardware.

6.1 Realism Supporting and Limiting Factor of the Effects

In this section, the factors that supported as well as limited the vibration realism of each visual effect are discussed. The order of the effects in the following section differs from the order in the previous figures and tables. We will first discuss the results of the baseline condition and afterward the results of the different tested VISUALEFFECTS in descending order, regarding their results.

6.1.1 SinusMotion. The VISUALEFFECT *sinusMotion* was used as our baseline condition, and the motion of the object was calculated by using a sinus function. This method is commonly used when implementing and presenting vibrations of objects; see Section 3. While this method achieved good results in terms of PERCEIVEDSTRENGTH, the results confirm the problem we aim to address with this work and show that this effect lacks REALISM in displaying a vibration. The well-perceived strength is also supported by the qualitative feedback, where the strength was mentioned as adequate for representing a vibration (*"The strength of the movement was optimal for vibration"*, P.18) and is in line with our findings within our pre-study in terms of defining the used amplitude; see Section 3. As a critique, participants mentioned the motion as erratic and flickering and not as smooth as you would imagine a movement of a vibration (*"Slight flickering, chopping"*, P.8) or (*"Very erratic movements of the object"*, P.9). This feedback confirms the findings of the technical analysis of the device (Section 3), as well as the findings of other works investigating the transfer of established methods onto AR HMDs [17, 29]. Established methods on other systems might need too many computational resources to be implemented in AR, or the devices' capabilities, like the functionality of the display and tracking, are not eligible [41]. This is supported by our findings, as even the transferring of basic mathematical functions results in a drop in frame rate and is not well perceived in terms of REALISM. Therefore, we also support the findings that other methods have to be found to simulate the required feedback and can be found with less computational power demanding approaches like it was done for motion blur [22] and other methods; see Section 2.

6.1.2 Object Blur. With our study, we were able to identify an animated blur effect applied to an AR object to represent a vibration of the displayed object significantly more realistically than the baseline. While the visualEffect *objectBlur* was perceived as strong as the baseline condition when presenting a vibrating object, it indeed was perceived as significantly more realistic than the baseline condition. The *objectBlur* mostly met the expectations of the participants of a vibration related to real-world experiences and how they believe a vibration would look like (*"The movement corresponded most closely to my idea of a vibrating object"*, P.18), or (*"Most like I imagine a vibration. No movement, but rather a change in place"*, P.19). Even though the object did not move, the *objectBlur* was able to simulate a vibrating movement of the object. A possible reason for being perceived as more realistic than other conditions might be that the motion of the *objectBlur* was felt as being faster than the baseline condition, as one participant mentioned (*"The vibration felt a bit faster hence more real"*, P.14). Especially, the limitations of computational power were repeatedly identified as the main reason that makes a transfer difficult or impossible; see Section 2. We can confirm and extend the findings that even when transferring less power-demanding methods to AR, alternative possibilities should be explored as they also here can improve the results, like in our work, the more realistic perception of vibrations. While this visualEffect was perceived best in terms of realism, it still can be increased and further investigated in the future. For example, the motion of the vibration was occasionally perceived as being too uniform overall for the presented strength (*"Too uniform for the vibration strength"*, P.13) or (*"For the strength, it looked too static"*, P.10).

6.1.3 Multiple Objects. Although the visualEffect *multipleObjects* was implemented as an extended method of the baseline, this effect produced comparable results to the baseline. No significant differences were found between *multipleObjects* and the baseline, neither for perceivedStrength nor for realism. A reason might be that both effects suffered from the same limiting factors, mainly regarding the presented motion; see Section 5. Even though more intermediate states were detected (*"There were more intermediate stages in this movement"*, P.17), a possible reason for this effect not being perceived more realistically than the baseline could be that the motion was still perceived as flickering and erratic, as one response suggests: (*"A bit too fast, slightly erratic. Frame rate too low?"*, P.8). This might result from the point that this effect also used a sinus function for the displacement of the objects, and the frame rate was again slightly around 40 Hz; see Section 3. This supports the findings of other research that adjusting an effect might slightly change the perceived effect, but is not strong enough to transfer the method adequate to AR [29]. In addition, the more intermediate stages mentioned by the participants have not necessarily led to improved realism. The aimed mimic of anti-aliasing could not improve the effect and might have brought new perception-based problems. For example, 2 participants mentioned that the resulting motion was perceived as more cartoony and less than a vibration movement (*"It felt a bit like it is moving rather than vibrating"*, P.14) and (*"It moved a little too much. Seemed very cartoony"*, P.20).

6.1.4 Background Blur. The visual effect that represented least realistically an object vibration was the *backgroundBlur*. Participants recognized that the background vibrated and not the object itself; see Section 5. Detecting an effect does not necessarily result in not perceiving the scene as intended. For example, in the case of depth of field or motion blur, see Section 3, it is also recognized that the background is blurred. In our study, however, participants might have found the movement of the background was a disruptive factor, as one participant mentioned: (*"By the fact that the* *background was vibrating, you could not determine whether the cube also vibrated"*, P.2). Hence, for presenting a vibrating object, moving a blurred background is not useful.

6.1.5 Design Recomendation. As long as most AR glasses do not have the ability to support anti-aliasing, we recommend using the visualEffect of an animated *objectBlur* to represent vibrating AR objects on devices such as HoloLens 2. While the presented study was conducted on the Hololens 2, the same recommendation can be made for other AR devices as the visualEffect of an animated *objectBlur*, based on a shader, is less power-demanding than the standard method of presenting a vibration calculating with a sinus function. Hence, using blurred edges for motion visualization can be generally recommended for AR headsets to save computing power and, for example, to extend the battery life. This will be beneficial, even if more powerful AR glasses with the computational ability to display anti-aliasing are available.

6.2 Benefitting Application Areas

Visually induced vibrations in AR could be helpful for use cases and application areas where haptics support interaction. It is well known that haptic feedback is a great support for gestures when selecting UI elements or exploring objects and surfaces. Controllers that could provide haptic feedback in AR are not very useful because they are not available or the use case keeps the hands busy, e.g., in assembly tasks or teaching and guiding manual skills. Our approach could support learning instruments and let the strings of the guitar swing realistically. UI elements in AR placed in midair could visually vibrate when being virtually touched. Furthermore, media that augments reality, such as virtual water and vibrating everyday surfaces of objects that are virtually shaken, could make Augmented Gaming much more immersive, realistic, and fun.

6.3 General Limitations And Future Work

One aspect that was mentioned occasionally for all the effects, in general, was the absence of sound. Being aware that sound could increase the realism of perceiving a vibration as the best-known property of vibration [7], we consciously decided to omit the use of sound in this work. The reason is that we wanted to investigate the differences in the effects purely visually and therefore eliminate all other influencing factors. Nevertheless, in future applications, a sound should be added when presenting a virtual vibration, as this would increase the overall experience. Regarding the sound, it should be explored how the visual representation of vibration and the corresponding sound could be matched best. Further, it might be of interest to investigate the *objectBlur* with having a texture on the blurred object. We only displayed an object of a single color (gray) without any other factors to avoid influencing the participants. Future investigations could also explore the blur effect on objects with texture or graphical objects that are related to vibrations, such as smartphones. In addition, it could be investigated whether *objectBlur* is perceived as a realistic motion for longer directed motion paths.

7 CONCLUSION

Inspired by the limitations of current state-of-the-art AR devices, it was explored how the perception of a visual vibration can be increased through different concepts while being mostly independent of the platform. Therefore, we used a HoloLens 2 device to show a gray virtual square on top of a virtual plane with a checkerboard texture to participants. Three effects (blurring edges, blurring background, moving multiple stacked objects) were explored and compared to a standard baseline condition (sinus wave for translation) and a control condition (no effect).

A user study showed that an edge blur effect with changing widths applied to a static object is perceived as significantly more realistic than the commonly used sinus motion used as a baseline. Qualitative feedback supports this finding. Hence, we conclude that the object's edge blur animation is a promising candidate to create the perception of a vibrating AR object and could be used as an alternative to the existing standard method.

ACKNOWLEDGMENTS

This project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 425869442 and is part of Priority Program SPP2199 Scalable Interaction Paradigms for Pervasive Computing Environments.

REFERENCES

[1] Timo Aila and Samuli Laine. 2009. Understanding the Efficiency of Ray Traversal on GPUs. In *Proceedings of the Conference on High Performance Graphics 2009* (New Orleans, Louisiana) *(HPG '09)*. Association for Computing Machinery, New York, NY, USA, 145–149. https://doi.org/10.1145/1572769.1572792

[2] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. 2019. *Real-time rendering*. AK Peters/crc Press, Abingdon, UK.

[3] Yuki Ban, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2012. Modifying an Identified Position of Edged Shapes Using Pseudo-Haptic Effects. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology* (Toronto, Ontario, Canada) *(VRST '12)*. Association for Computing Machinery, New York, NY, USA, 93–96. https://doi.org/10.1145/2407336.2407353

[4] Joanna Bergström, Aske Mottelson, and Jarrod Knibbe. 2019. Resized Grasping in VR: Estimating Thresholds for Object Discrimination. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1175–1183. https://doi.org/10.1145/3332165.3347939

[5] James V Bradley. 1958. Complete counterbalancing of immediate sequential effects in a Latin square design. *J. Amer. Statist. Assoc.* 53, 282 (1958), 525–528.

[6] Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*. Association for Computing Machinery, New York, NY, USA, 137–145. https://doi.org/10.1145/800031.808590

[7] Gautier Dagan, Frank Keller, and Alex Lascarides. 2023. Learning the Effects of Physical Actions in a Multi-modal Environment. arXiv:2301.11845 [cs.CL]

[8] Grigoris Daskalogrigorakis, Ann McNamara, and Katerina Mania. 2021. Holo-Box: Level-of-Detail Glanceable Interfaces for Augmented Reality. In *ACM SIGGRAPH 2021 Posters* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. https://doi.org/10.1145/3450618.3469175

[9] Eleanor J Gibson, James J Gibson, Olin W Smith, and Howard Flock. 1959. Motion parallax as a determinant of perceived depth. *Journal of experimental psychology* 58, 1 (1959), 40.

[10] Andrew S Glassner. 1989. *An introduction to ray tracing*. Morgan Kaufmann, Massachusetts, USA.

[11] Brian Gleeson and David E Johnson. 2010. Expressive haptic rendering with cartoon-inspired effects. In *2010 IEEE Haptics Symposium*. IEEE, IEEE, Waltham, MA, USA, 191–194.

[12] Taku Hachisu, Gabriel Cirio, Maud Marchal, Anatole Lécuyer, and Hiroyuki Kajimoto. 2011. Pseudo-haptic feedback augmented with visual and tactile vibrations. In *2011 IEEE International Symposium on VR Innovation*. IEEE, IEEE, Singapore, 327–328.

[13] Taku Hachisu, Gabriel Cirio, Maud Marchal, Anatole Lécuyer, and Hiroyuki Kajimoto. 2015. *Visual Vibrations to Simulate Taps on Different Materials*. Springer Japan, Tokyo, 297–303. https://doi.org/10.1007/978-4-431-55690-9_54

[14] Tan Kim Heok and D. Daman. 2004. A review on level of detail. In *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.* IEEE, Penang, Malaysia, 70–75. https://doi.org/10.1109/CGIV.2004.1323963

[15] Namkyoo Kang, Young June Sah, and Sangwon Lee. 2021. Effects of visual and auditory cues on haptic illusions for active and passive touches in mixed reality. *International Journal of Human-Computer Studies* 150 (2021), 102613.

[16] Y. Kawagishi, K. Hatsuyama, and K. Kondo. 2003. Cartoon blur: nonphotorealistic motion blur. In *Proceedings Computer Graphics International 2003*. IEEE, Tokyo, Japan, 276–281. https://doi.org/10.1109/CGI.2003.1214482

[17] Jonghyun Kim, Youngmo Jeong, Michael Stengel, Kaan Akşit, Rachel Albert, Ben Boudaoud, Trey Greer, Joohwan Kim, Ward Lopes, Zander Majercik, Peter Shirley, Josef Spjut, Morgan McGuire, and David Luebke. 2019. Foveated AR: Dynamically-Foveated Augmented Reality Display. *ACM Trans. Graph.* 38, 4, Article 99 (jul 2019), 15 pages. https://doi.org/10.1145/3306346.3322987

[18] Microsoft. 2022. Understanding Performance for Mixed Reality - Mixed Reality | Microsoft Learn. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/understanding-performance-for-mixed-reality. (Accessed on 03/13/2023).

[19] Bipul Mohanto, ABM Tariqul Islam, Enrico Gobbetti, and Oliver Staadt. 2022. An integrative view of foveated rendering. *Computers & Graphics* 102 (2022), 474–501. https://doi.org/10.1016/j.cag.2021.10.010

[20] Fernando Navarro, Francisco J Serón, and Diego Gutierrez. 2011. Motion blur rendering: State of the art. In *Computer Graphics Forum*. Wiley Online Library, Wiley Online Library, New Jersey, US, 3–26.

[21] Forrest W Nutter Jr. 2010. Weber-Fechner Law. *Encyclopedia of research design* 3 (2010), 1612–1615.

[22] Youngmin Park, Vincent Lepetit, and Woontack Woo. 2011. Handling motion-blur in 3d tracking and rendering for augmented reality. *IEEE transactions on visualization and computer graphics* 18, 9 (2011), 1449–1459.

[23] Michael Rietzler, Florian Geiselhart, Julian Frommel, and Enrico Rukzio. 2018. Conveying the Perception of Kinesthetic Feedback in Virtual Reality Using State-of-the-Art Hardware. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174034

[24] Michael Rietzler, Gabriel Haas, Thomas Dreja, Florian Geiselhart, and Enrico Rukzio. 2019. Virtual Muscle Force: Communicating Kinesthetic Forces Through Pseudo-Haptic Feedback and Muscle Input. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 913–922. https://doi.org/10.1145/3332165.3347871

[25] Matt Ritchie, Greg Modern, and Kenny Mitchell. 2010. Split Second Motion Blur. In *ACM SIGGRAPH 2010 Talks* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 17, 1 pages. https://doi.org/10.1145/1837026.1837048

[26] Brian Rogers and Maureen Graham. 1979. Motion parallax as an independent cue for depth perception. *Perception* 8, 2 (1979), 125–134.

[27] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare Parise. 2019. Pseudo-Haptic Weight: Changing the Perceived Weight of Virtual Objects By Manipulating Control-Display Ratio. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300550

[28] Artur Santos, João Marcelo Teixeira, Thiago Farias, Veronica Teichrieb, and Judith Kelner. 2012. Understanding the efficiency of KD-tree ray-traversal techniques over a GPGPU architecture. *International Journal of Parallel Programming* 40 (2012), 331–352.

[29] Artur Lira Dos Santos, Diego Lemos, Jorge Eduardo Falcão Lindoso, and Veronica Teichrieb. 2012. Real Time Ray Tracing for Augmented Reality. In *2012 14th Symposium on Virtual and Augmented Reality*. IEEE, Rio de Janeiro, Brazil, 131–140. https://doi.org/10.1109/SVR.2012.8

[30] Takeharu Seno and Haruaki Fukuda. 2012. Stimulus meanings alter illusory self-motion (vection)—experimental examination of the train illusion. *Seeing and perceiving* 25, 6 (2012), 631–645.

[31] Ana Serrano, Incheol Kim, Zhili Chen, Stephen DiVerdi, Diego Gutierrez, Aaron Hertzmann, and Belen Masia. 2019. Motion parallax for 360 RGBD video. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 1817–1827.

[32] Tiago Sousa. 2008. Crysis next gen effects. In *Game Developers Conference*, Vol. 2008. Game Developers Conference, London, UK, 45–67.

[33] GH Spencer and MVRK Murty. 1962. General ray-tracing procedure. *JOSA* 52, 6 (1962), 672–678.

[34] Stanley S Stevens. 1960. The psychophysics of sensory function. *American scientist* 48, 2 (1960), 226–253.

[35] Patrick L. Strandholt, Oana A. Dogaru, Niels C. Nilsson, Rolf Nordahl, and Stefania Serafin. 2020. Knock on Wood: Combining Redirected Touching and Physical Props for Tool-Based Interaction in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376303

[36] Anselm Strauss and Juliet M Corbin. 1997. *Grounded theory in practice*. Sage, California, US.

[37] R A Syaputra and A Widiyanto. 2020. Implementation of the level of detail method on augmented reality android-based applications. *Journal of Physics: Conference Series* 1517, 1 (apr 2020), 012090. https://doi.org/10.1088/1742-6596/1517/1/012090

[38] Léo Terziman, Maud Marchal, Franck Multon, Bruno Arnaldi, and Anatole Lécuyer. 2012. The King-Kong Effects: Improving sensation of walking in VR with visual and tactile vibrations at each step. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, IEEE, Costa Mesa, CA, USA, 19–26.

[39] Alex Vlachos. 2008. Post processing in the orange box. In *Game Developers Conference*, Vol. 1. Game Developers Conference, London, UK, 45–67.

[40] Lili Wang, Xuehuai Shi, and Yi Liu. 2023. Foveated rendering: A state-of-the-art survey. *Computational Visual Media* 9, 2 (2023), 195–228.

[41] Matthias M. Wloka and Brian G. Anderson. 1995. Resolving Occlusion in Augmented Reality. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics* (Monterey, California, USA) *(I3D '95)*. Association for Computing Machinery, New York, NY, USA, 5–12. https://doi.org/10.1145/199404.199405

[42] Katrin Wolf and Timm Bäder. 2015. Illusion of Surface Changes Induced by Tactile and Visual Touch Feedback. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. Association for Computing Machinery, New York, NY, USA, 1355–1360. https://doi.org/10.1145/2702613.2732703

[43] Sung-eui Yoon, Enrico Gobbetti, David Kasik, and Dinesh Manocha. 2008. Real-time massive model rendering. *Synthesis Lectures on Computer Graphics and Animation* 2, 1 (2008), 1–122.