

On Virtual Pointing Rays and Motion Artefacts

Gareth Rendle*

Adrian Kreskowski†

Bernd Froehlich‡

Virtual Reality and Visualization Research Group
Bauhaus-Universität Weimar
Weimar, Germany

ABSTRACT

Virtual pointing rays are a commonly employed interaction metaphor allowing users of mixed reality applications to select out-of-reach virtual objects. However, fast hand movements can cause the visual representation of the ray to move quickly across the visual field, leading to motion artefacts that cause multiple copies of the ray to appear in discrete locations. In this work, we conduct a user study to investigate whether the conditions for motion artefact occurrence arise during a selection task that participants undertake with a pointing ray. We analyse the recorded gaze and ray movement data from the study to show that pointing actions create the conditions necessary for motion artefacts to appear. In addition, we propose a motion blur effect that we apply to the pointing ray with the aim of mitigating motion artefacts. The blur effect is evaluated in the aforementioned user study and is found to improve the perceived smoothness of ray selection, without affecting selection performance.

Index Terms: Pointing Ray, Virtual Reality, Motion Artifacts

1 INTRODUCTION

Virtual pointing rays enable users of Virtual Reality (VR) applications to interact with the virtual environment. Pointing rays typically follow a ‘laser beam’ metaphor, appearing as a beam of light projecting away from the user’s hand. Their potentially infinite reach allows selection of distant virtual objects [28], interaction with user interface menus, and specification of the desired destination during navigation [5]. While variations of the pointing ray have been suggested to address limitations that arise in certain use cases [1, 2, 14], its low implementation overhead and comprehensible nature mean that the simple pointing ray remains a commonly employed interaction technique.

One negative side-effect of employing pointing rays in VR applications that has not been explicitly addressed in the literature is the occurrence of motion artefacts when the pointing ray moves quickly across the visual field. Typically, the position and orientation of a pointing ray are controlled directly by the position and orientation of the user’s hand (or handheld controller). Fast hand movements during selection or navigation cause the visual representation of the ray displayed on the user’s Head-Mounted Display (HMD) to move quickly across the visual field. These motions can lead to the appearance of *multiple image artefacts*, which have been shown to occur when the displacement of the stimulus between consecutive frames shown by the display exceeds a certain threshold [4]. Since motion artefacts negatively affect the perceived quality of motion [24], techniques for motion artefact mitigation are likely to lead to an improved user experience.

*e-mail: gareth.rendle@uni-weimar.de

†e-mail: adrian.kreskowski@uni-weimar.de

‡e-mail: bernd.froehlich@uni-weimar.de

In this work, we investigate whether the conditions for motion artefact occurrence arise during a pointing ray selection task in VR. We ground our investigation in previous work that identifies when motion artefacts appear. The paper is based on a user study in which participants perform a standardised selection task with a pointing ray. We analyse gaze and movement tracking data generated during the study to assess whether pointing actions in our study scenario are likely to lead to the perception of motion artefacts. To mitigate the appearance of motion artefacts, we implement a motion blur effect for simple pointing rays that addresses the causes of multiple imaging artefacts. The impact of the motion effect on selection quality and motion artefact perception is tested in the aforementioned user study.

Our analysis of relative gaze and ray movement indicates that the conditions required for motion artefacts to appear do occur while selecting objects with a virtual pointing ray. The user study finds that our motion blur effect increases the perceived smoothness of selection, and did not provide any indication that participants’ selection performance was negatively affected. The motion blur was therefore found to have a positive overall effect on user experience for ray selection tasks in VR.

2 BACKGROUND AND RELATED WORK

Motion artefacts are imperfections that may be perceived when the impression of motion is created by displaying a sequence of discrete static images. In this section, we give an overview of the types of motion artefacts, which can be differentiated by their appearance and categorised by the relative movement between the eye and stimulus when they occur. All of the artefact types discussed are displayed in Figure 1. Subsequently, we highlight how virtual pointing rays can cause motion artefacts to appear, before discussing literature describing techniques for mitigating motion artefacts.

2.1 Motion Artefacts: No Relative Eye/Stimulus Motion

If both the stimulus and the eye are static, motion artefacts will not appear. *Flicker*, however, is sometimes categorized as a motion artefact (e.g. [11]) and occurs if the luminance of a stimulus fluctuates at a frequency lower than the Critical Flicker Frequency (CFF), a threshold at which the visual system can detect luminance changes (typically reported as 50-90 Hz [10]). Since modern HMDs have a refresh rate above 60 Hz, flicker caused purely by luminance fluctuations is rarely an issue in VR applications.

Stimuli moving at speeds up to 80 deg/s can be tracked by Smooth Pursuit Eye Movements (SPEM) [8], resulting in the retinal image of a stimulus being stabilized within the fovea (although perfect stability is not achieved in practice). When the eye tracks stimuli moving on a display, motion artefacts can appear due to the differences between the continuous motion of the eye and the discrete motion of the stimulus on the display. *Judder* (or *non-smooth motion*) appears because the temporally quantised motion of the stimulus deviates from the smooth trajectory expected by SPEM [9, 21], and occurs even at low movement speeds. Judder-induced *edge flicker* occurs when the edges of the object do not

appear at a stable position on the retina; as the eye moves continuously to track a stimulus, each discrete frame results in the stimulus being 'smeared' across the retina, resulting in a flickering effect caused by the unstable retinal position of the stimulus' edges [22]. On high-persistence displays, this smear results in *motion blur* [39], which is why modern HMDs use low-persistence displays.

When a moving stimulus is displayed at the same location for multiple frames (e.g. when the update rate of the stimulus is lower than the refresh rate of the display), multiple visible copies of the stimulus may appear in distinct locations, again because the continuous tracking movement of the eye means that each repetition of the stimulus appears at a different position on the retina. This artefact is most visible at the edges of the stimulus and is therefore referred to as *edge banding* [19].

2.2 Motion Artefacts: Relative Eye/Stimulus Motion

Fast relative motion between eye and stimulus occurs either when a stimulus moves without being tracked by the eye, or when the eye moves rapidly (i.e. saccades) without tracking a stimulus. In both cases, the retinal position of the stimulus can change rapidly, resulting in motion artefacts. During saccadic eye motion, stimuli exhibiting fast luminance changes (or *strobing*), even at frequencies above 1kHz, can cause *phantom arrays* to be perceived: multiple copies of the stimulus that appear as a trail extending away from the stimulus in the direction of the saccade [17, 34]. Research has suggested that saccadic eye movements allow the detection of flicker on displays showing stimuli with high spatial frequencies at much higher rates than the typically reported CFF [10].

When the stimulus moves with respect to the eye, for example when the gaze is fixated on a static point while a stimulus moves, multiple copies of the stimulus may be visible. This phenomenon can be demonstrated by viewing a spinning wheel marked with a radial line that is illuminated by a strobe light. A trail of lines will appear to follow the actual stimulus [7]. On a video display, this *multiple image* artefact was found to occur when the displacement undergone by the stimulus on the retinal image between two consecutive frames exceeds a certain threshold [4]. If the refresh rate of a display is fixed, then faster motion results in larger inter-frame displacements, meaning that fast-moving objects are more likely to cause multiple image artefacts.

2.3 Pointing Rays and Multiple Image Artefacts

Several factors outlined in previous works provide some explanation of why virtual pointing rays are susceptible to multiple image artefacts. Firstly, research on directed pointing in both the real world and VR has observed that pointing motions begin with an initial ballistic motion of the pointer towards the target, before a correction phase where smaller adjustments occur based on visual feedback [43, 23]. If the user's gaze fixates on the target while moving the ray, rather than tracking the ray as it moves towards the target, this would cause the retinal position of the ray to move quickly. The initial ballistic movement of the ray may be guided by proprioception instead of visual feedback. If the inter-frame displacement of the ray is large enough, multiple image artefacts will occur, as outlined by Bex et al. [4].

In addition to the movement of a pointing ray, the typical visual representation of the ray is conducive to multiple image artefacts. If the ray appears as a beam with defined edges, it is a stimulus with high spatial frequencies. Bex et al. [4] found that stimuli with higher spatial frequencies have a lower inter-frame displacement threshold at which multiple image artefacts occur than low-pass or band-pass filtered stimuli; meaning that, if a fixed frame rate is assumed, lower movement speeds are required for stimuli with high spatial frequencies to cause artefacts.

Furthermore, the visual representation of the ray must be visible against the background of the virtual scene. The ray is therefore

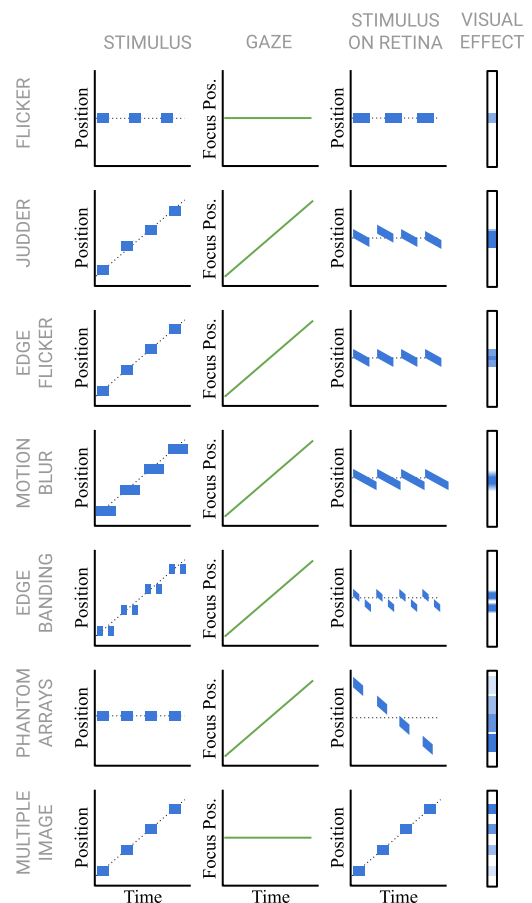


Figure 1: Types of motion artefacts. The first column shows stimulus activation and movement relative to the viewer. The second column shows the movement of the viewer's eye. The third column shows the retinal position of the stimulus that results from the relative stimulus/eye movement. The final column represents the visual effect that occurs at the end of the time axis. Note that this figure is intended to explain differences between types of motion artefacts, and does not display experimental data. Adapted and extended from images in [19, 22].

likely to be brightly coloured to stand out against the scene, meaning that the moving stimulus has high luminance and high contrast (where contrast is defined as variation in luminance, in this case between the ray and the background). Both contrast and luminance affect the visibility of temporal aliasing artefacts [39].

While using displays with a higher refresh rate would help to alleviate multiple image artefacts (due to the lower inter-frame displacement of moving stimuli, when compared with lower refresh rate displays), this would require a significant increase in the required computational resources to render the scene.

2.4 Reducing Motion Artefacts

Research on the perception of moving objects in the real world has shown that motion causes the observer to perceive motion blur that depends on the object's retinal velocity [31]. When the object is tracked with SPEM, a deblurring effect occurs [6]. However, in the case of a pointing ray, we expect the gaze to focus on the selection target rather than tracking the movement of the ray, foregoing the deblurring effect. We investigate the relative movement of gaze and ray in Section 5.

Viewers are also accustomed to seeing motion blur when moving stimuli are recorded with film and electronic cameras. The camera's shutter opens for a finite duration, during which light from the range of positions occupied by a stimulus is integrated into the recorded image, producing a blurred impression of the stimulus that is 'smeared' in the motion direction. Since images created with computer graphics render the scene state at a single instant in time, they must explicitly simulate motion blur, a challenge that has been researched for decades for the dual purpose of replicating the effect of physical cameras and reducing motion artefacts (also known as *anti-aliasing*). We note that although the term *temporal anti-aliasing* initially referred to techniques that reduced temporal aliasing artefacts (e.g. [20, 15]), it is nowadays more commonly used to refer to techniques that employ multiple temporal samples to combat spatial aliasing artefacts, as evidenced by a recent survey [44]. As such, we avoid the term in this work.

Numerous methods have been proposed to simulate motion blur in computer-generated images [30], including recent methods capable of real-time motion blur rendering [27, 16, 25]. In principle, motion blur rendering algorithms aim to determine pixel colour by integrating colour contributions from each visible scene object during a given interval [38]. Some methods perform spatiotemporal super-sampling, leveraging stochastic sampling to reduce the number of samples required [13]. Analytical solutions can be evaluated exactly, but rely on assumptions about object motion [20]. Some approaches substitute the geometry of an object for an adaptation that accounts for its evolution over time [32, 15, 25]. While many methods aim to render physically accurate blur effects, some do not; instead producing 'motion hints' that may be abstract, artistic indicators of motion, or approximations of blur that cannot easily be distinguished from physically accurate blur [41, 36, 29]. In the context of fast mouse cursor movement, a motion effect that creates additional, distinct versions of the cursor was proposed to help users follow its trajectory [3].

Some commercial VR games that involve fast movements of swords and sabers, such as Beat Saber¹ and Ironlights², apply exaggerated motion trails to the handheld objects. In this context, such effects may serve a dual purpose, creating a stylized visual experience while reducing motion artefacts. Applying a motion blur effect to the moving ray or saber reduces the amount of high spatial frequencies. Low-pass filtered stimuli are known to be less likely to cause multiple image artefacts [4]. So, while application cases can be found in the VR industry, no published work has explicitly investigated the effect of applying motion blur to either sabers or virtual pointing rays.

3 MOTION BLUR EFFECT

Virtual pointing rays are susceptible to multiple image artefacts for the reasons outlined in Section 2.3. In this work, we investigate the effect of applying an anti-aliasing motion blur to the ray. Our motion blur effect is based on motion blur approximation techniques described in the literature [36, 29, 41], which we apply to our specific use case of a hand-driven virtual pointing ray. This section details the design and implementation of the motion blur effect.

3.1 Preliminaries

The ray is attached to one of the user's tracked controllers. The controller's position and orientation define the start point of the ray R , the ray direction \mathbf{v} and an orthogonal upwards vector \mathbf{a} . A single eye position E is the centroid of the positions of the left and right eye. The non-blurred visual representation of the pointing ray consists of a long quadrilateral with one short side intersecting R , extending in direction \mathbf{v} , with surface normal \mathbf{n} equal to \mathbf{a} . The quad

is formed in practice by two triangles. To maintain a constant ray width from the user's point of view, the ray quad is rotated around \mathbf{v} at each frame such that \mathbf{v} , \mathbf{n} , and $\mathbf{e} = RE$ lie on a plane. The ray is shaded with a fixed colour, disregarding illumination; this is consistent with the laser beam metaphor followed by many pointing ray implementations and simplifies motion blur calculations.

3.2 Design

Analytical solutions for motion blur have been proposed that, under the assumption of linear motion, calculate which parts of an object project to a pixel during a given time interval [36]. When applying blur to a single object that has a constant colour, the proportion of the time interval during which the object projects onto a pixel can be used to determine its opacity in a pixel shader program. The objects' opacity is inherently reduced when undergoing movement, since it 'covers' each pixel for a shorter proportion of the time interval. High spatial frequencies are also reduced when movement occurs, as adjacent pixels are covered by the object for similar proportions of the time interval.

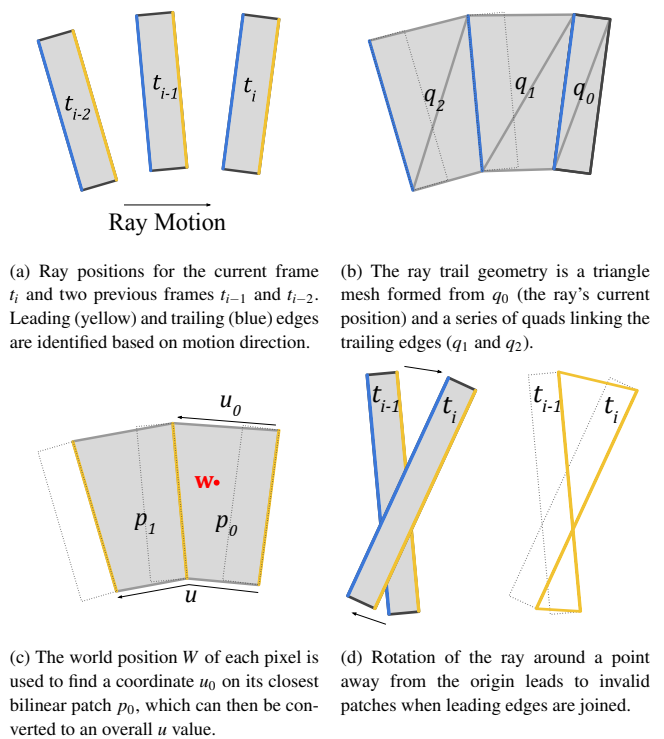


Figure 2: Illustration of the steps required to implement our motion blur approximation effect for ray geometry. A fixed number of ray positions across subsequent frames (a) is connected by geometry representing the ray's trail over time (b). Using bilinear interpolation across this geometry (c) allows for a per-pixel computation of ray opacity, resulting in a smooth motion blur effect. The geometry is a suitable representation of the ray's trail over time, except when the ray is rotated around a point forward of the ray's origin (d).

While reducing stimulus opacity (and therefore luminance) and high spatial frequencies are beneficial for avoiding motion artefacts, our initial tests using an analytical approach to generate motion blur for the time interval between the current and previous frames (t_i and t_{i-1}) still yielded multiple image artefacts, since the blurred version of the ray still appeared in distinct locations at each frame. This could be mitigated to some degree by calculating blur over multiple frames (e.g. from t_{i-4} to t_i), resulting in a longer trail behind

¹<https://beatsaber.com/>

²<https://www.ironlightsgame.com/>

the ray. However, sections of the ray trail formed between each pair of time steps could have different opacity values (depending on the movement speed during the interval), producing a trail that appeared to be temporally unstable as the ray moved. To avoid this temporal instability, we deviate from the analytical approach, instead favouring a blur approximation that varies opacity smoothly along the ray trail.

To create the ray trail, we take inspiration from previous motion blur approximation approaches that substitute the geometry of the moving object with an object that represents the object's path over time (e.g. lines to represent moving points [32] or capsules representing moving spheres [29]). We transfer this idea to the context of a moving ray, creating a geometric representation of the ray's positions during multiple frames, up to and including the current frame t_i (Figure 2a), by joining the quad corresponding to the ray's position at t_i with a set of quads that join the ray's trailing edges from the previous N frames (Figure 2b). The trailing edge is identified by transforming the last inter-frame movement vector of the ray centroid into camera space. If the vector moves right (positive x component) then the left edge is the trailing edge. The resulting set of quads is rendered as a triangle mesh, where each quad is represented by a pair of triangles.

The opacity of each pixel of the ray trail mesh is determined in the pixel shading stage. For each pixel on the ray trail with world space W , a normalized coordinate u along the trail is calculated. To find u , bilinear patches are formed between temporal instances of the ray's leading edge (Figure 2c). An intersection test between the eye-to-pixel vector $w = EW$ and each bilinear patch, using the approach described by Reshetov [33], yields a u coordinate for each pixel that varies smoothly along each section of the trail (note that u would not vary smoothly if it was derived using barycentric coordinates of the triangles that form a quad between leading edge instances). The function $\alpha(u)$, shown in Figure 3a, determines the position-dependent opacity of each pixel on the trail (similar to the ramp function used in [41]), and is multiplied by the result of the speed-dependent opacity function $\alpha(s)$ (Figure 3b), similar to the approach of Tatarchuk et al. [29]. The position-dependent opacity function creates an approximation of motion blur, in that it avoids defined edges by having smooth changes in opacity. The dependency of opacity on speed reflects the result of physically based motion blur effects, since pixels on the trail are covered for a shorter time by the ray when the ray moves faster, meaning that the ray's contribution to the colour of the pixel is lower (as shown in the second image of Figure 4b, ray moving at 30 m/s). To enable a smooth degradation to a non-blurred ray as ray movement speed decreases, a linear transition between a fully opaque ray and the blurred ray is also implemented. A comparison of selection actions with and without the motion blur effect can be seen in Figure 4.

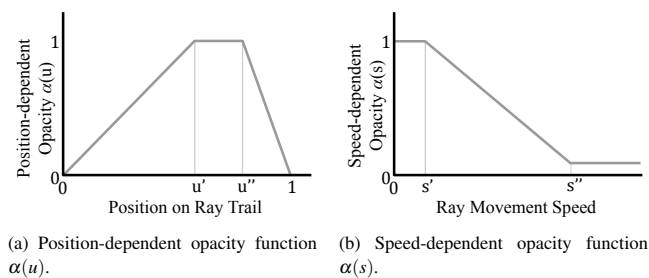


Figure 3: Opacity functions that vary based on the pixel's position on the ray trail (a) and the ray's speed (b). The opacity values obtained from the transfer functions are multiplied to determine the final opacity of each pixel on the ray trail.

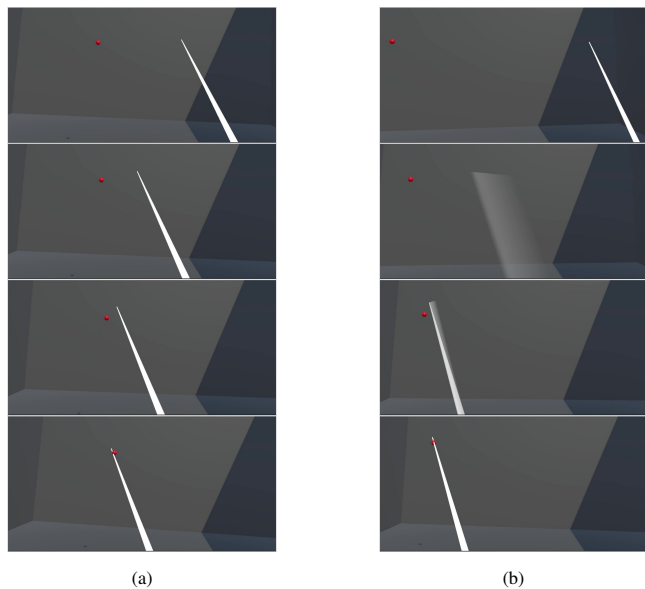


Figure 4: Comparison of ray appearance during selection without (a) and with (b) the motion blur effect.

3.3 Implementation Details

The motion blur effect was implemented in Unity 2021.3.21f1. The vertex positions for the ray trail's triangle mesh are updated at each frame from a C# script, based on the previous N positions of the ray and the ray movement direction. The ray trail geometry is rendered as a Unity mesh with a transparency-enabled shader, allowing Unity to handle transparency correctly. The implementation is available in the online repository associated with this paper³. The motion blur effect has a negligible impact on performance, since it only requires a small triangle mesh to be rendered.

3.4 Motion Blur Effect: Limitations

We designed our blur approximation for a subset of pointing rays that can be implemented as a single, fixed-colour quad. It is therefore not comprehensive, so does not support other geometric representations of the ray (e.g. parabolas) or rays with varying colour. We also note that the quad-based concept of a ray trail fails when the ray rotates around a point forward of the user's hand (see Figure 2d). We handle this case by applying a linear transition between the blurred and non-blurred versions of the ray near the hand, hiding artefacts caused by rotation around points forward of the hand.

4 USER STUDY

To fulfil the dual purpose of evaluating the motion blur effect introduced in Section 3 and collecting gaze and ray tracking data during a selection scenario, we conducted a user study in which participants performed a selection task with a virtual ray. Participants rated qualitative aspects of the pointing interaction and reported the perception and impact of motion artefacts.

4.1 Pointing Task

Participants undertook a VR version of the multi-directional tapping task [18] which was also used in recent VR pointing studies [14, 45, 35, 46]. During each trial, 9 spherical targets appeared sequentially in front of the participant. The target positions were equally spaced around a circle with radius r , as shown in Figure 5a. Consecutive targets always appeared on the opposite side of the

³<https://github.com/vrsys/RayMotionBlurEffect>

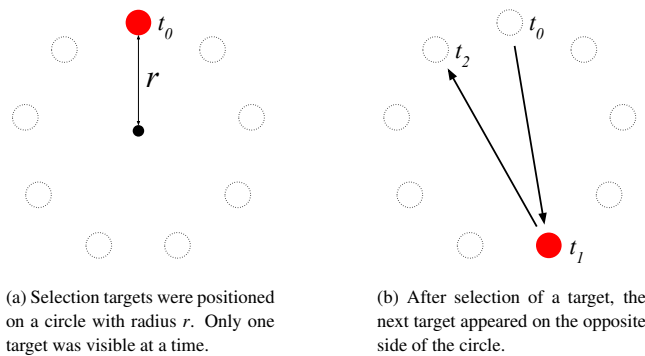


Figure 5: Positioning (a) and appearance order (b) of targets for the selection task.

circle, meaning that the inter-target distance was constant within each trial (Figure 5b). To make the sequence less predictable, the starting target and the order of progression through the targets (i.e. whether the next target was four or five targets around the circle in a clockwise manner) were randomised.

To complete the task, participants had to select each target when it became visible. Selection was indicated by continuously intersecting the target with the ray for one second, avoiding the need for confirmation through a button press. A short sound was played to confirm each selection. The study was performed while seated on a swivel chair. Participants chose whether to control the ray with their left or right hand.

4.2 Independent Variables

Four independent variables were investigated during the experiment, each with two levels. The independent variables were:

- *blur mode*: presence of the motion blur effect described in Section 3 (on/off);
- *display refresh rate* of the HMD (72 fps/90 fps);
- *target position radius*: radius r of the circle on which targets were positioned (1m/2m);
- *priming*: whether the trial took place before or after the participant was given information about the type of motion artefacts that may occur when using a virtual pointing ray (before/after).

The blur mode was chosen as an independent variable to evaluate how the blur effect impacted user experience and whether it reduced the perception of motion artefacts. The display refresh rate was included as an independent variable to assess its relative importance compared to the motion blur effect. Varying the target position radius allowed us to determine whether the impact of the motion blur effect applies to pointing tasks that require different magnitudes of motion, since target radius values of 1m and 2m result in a required ray movement angle of 28 and 52 degrees, respectively.

Priming was included as an independent variable because various existing works on motion artefacts take the approach of informing participants about potentially visible motion artefacts before the study [26, 9, 37]. We aimed to examine the effect of priming on participants' detection of motion artefacts to assess how priming participants at the start of the study might have affected results from previous studies.

4.3 Dependent Variables

As well as selection performance (i.e. time from target appearance to selection), we recorded the position and orientation of the ray, and the eye position and gaze direction during each trial.

After each trial, participants were asked to rate the following qualities by indicating their agreement with a related statement on a scale from 1 to 7:

- smoothness (“selecting objects with the ray felt smooth”);
- shakiness (“selecting objects with the ray felt shaky”);
- motion artefact severity (“I noticed some motion artefacts”);
- distraction caused by artefacts (“I felt distracted from the pointing task by motion artefacts”).

4.4 Study Design

Procedure. Participants were asked to read and sign a consent form on arrival. They were informed that gaze and hand position would be tracked and recorded, but would not be made publicly available. They completed a Snellen test for visual acuity and were given a demographic questionnaire. They were then briefed on the structure of the experiment.

Experiment Structure. The main experiment was split into four blocks, for two reasons. Firstly, half of the trials must take place before priming (P_B), and half after (P_A). This meant that a break in the middle of the experiment was required to inform participants about motion artefacts. Secondly, it was not possible to change the refresh rate of the HMD programmatically on a per-trial basis, instead requiring manual adjustment. To minimize wait time while the adjustment was made, trials in each half of the experiment were split again into two blocks, each of which was performed with a different display refresh rate (R_{72} or R_{90}). The block order therefore has four permutations:

1. $P_B R_{72} \rightarrow P_B R_{90} \rightarrow P_A R_{72} \rightarrow P_A R_{90}$
2. $P_B R_{90} \rightarrow P_B R_{72} \rightarrow P_A R_{72} \rightarrow P_A R_{90}$
3. $P_B R_{72} \rightarrow P_B R_{90} \rightarrow P_A R_{90} \rightarrow P_A R_{72}$
4. $P_B R_{90} \rightarrow P_B R_{72} \rightarrow P_A R_{90} \rightarrow P_A R_{72}$

The block order was partially counterbalanced. Due to an odd number of participants, permutation 4 was performed by three participants, while other permutations were performed by four participants. Within a block, the blur mode (motion blur effect on or off) and target position radius (1m or 2m) varied, giving four possible conditions. Each condition was performed twice per block, meaning that each block consisted of 8 trials. The order of the 8 trials within each block was randomised.

Priming. After the second block, participants were presented with a short presentation which verbally and visually introduced the concept of multiple image artefacts. The presentation included a short video showing a fast-moving stimulus that caused multiple image artefacts to appear, and explained that such artefacts can occur when moving the pointing ray during selection.

Blur Mode Preference. After the fourth block, participants were asked to directly compare the blur modes and to indicate their preference and the reasons for their choice. Participants were able to switch between modes using a button on their tracked controller while performing the same selection task as in the main study. Participants were permitted to indicate no preference.

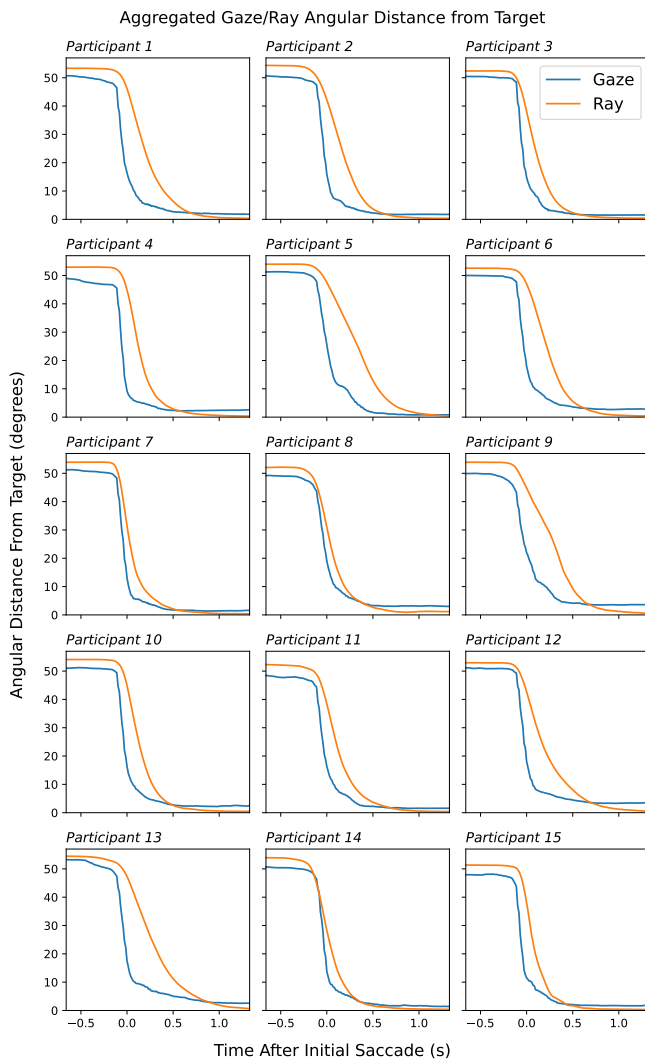


Figure 6: Aggregated angular distance from the target for the gaze and ray of each participant during selection motions.

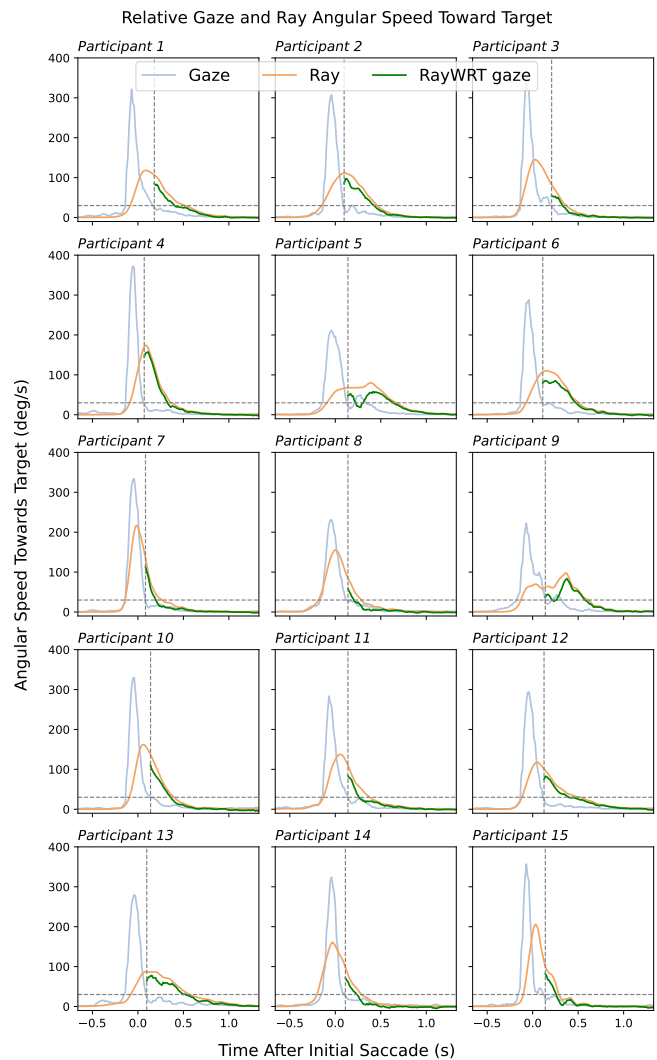


Figure 7: Angular speed towards the target, derived from the aggregated gaze and ray movements for each participant during a selection motion. Ray movement speed relative to the gaze is shown in green, starting from the end of the gaze's saccade towards the target.

5 EVALUATION

15 participants (10 male and 5 female) completed the user study described in Section 4. They were aged 23 to 34 (mean=25.8, SD=2.88). As the study was conducted at a university without an established ethics committee, no ethical approval was sought. While four participants reported that they used VR technology on a weekly basis, seven used VR on a monthly or annual basis, and three had never experienced VR. Five participants played computer games on a weekly or daily basis, while the remaining ten played computer games on a monthly or annual basis.

The HMD used was the Meta Quest Pro, enabling the collection of eye tracking data⁴. A recent study [40] reports an average accuracy of 1.7 degrees. We recorded a single gaze direction and eye position that were averaged from the respective gaze directions and eye positions of the left and right eyes.

The HMD was running in Quest Link mode, connected by USB

⁴<https://developer.oculus.com/documentation/unity/move-eye-tracking/>

cable to a PC with an Intel Core i9-13900K processor operating at 3.00 GHz, 128 GB of RAM, and an NVIDIA GeForce RTX 4090 graphics card.

5.1 Ray and Gaze Tracking

5.1.1 Relative Movement Towards Target

We aim to investigate the relative movement of the ray and gaze during a typical selection motion. Accordingly, we split tracking data from each trial into motions, with each motion initially ending at the time frame when the ray first intersects a target. To simplify the analysis of two streams of 6 DOF tracking data, we derive the per-frame angular distance from the target from each stream; that is, the rotation of the ray or gaze around the current eye or hand position that is required to point the ray/gaze at the centre of the target.

To enable meaningful aggregation of the angular motion paths from individual motions, it is necessary to align the motions temporally. We observed that many motions began with a fast eye movement (saccade) that covered most of the angular distance between

targets. The time frame corresponding to the start of the initial saccade was identified and used to align motions.

The initial saccade was detected by calculating a moving average of angular movement speed for the gaze during each motion and identifying when the maximum speed value occurred. If the saccade associated with this time point covered more than 60 % of the total angular distance between targets, it was accepted as a saccade. Motions without an accepted saccade were excluded from the analysis, leaving 70% of the originally included selection motions.

After aligning motions, the average angular distance from the target at each time frame could be calculated for the ray and the gaze. We observed significant differences between participants, so present the averaged ray and gaze paths separately for each participant in Figure 6.

We note that to facilitate aggregation, only trials where the target radius r was 2 m were included in the analysis. Additionally, trials recorded with a refresh rate of 90 fps were downsampled to 72 fps.

5.1.2 Relative Gaze-Ray Movement Speed

Saccade-based alignment of motions, used to calculate the aggregated movement paths, was also used to aggregate the speed of the gaze and the ray towards the target during a selection motion. For each motion, the per-frame angular speed towards the target was calculated for the gaze and the ray, and smoothed using a 5-frame moving average. The relative speed of the ray with respect to the gaze was also calculated for each motion on a per-frame basis. The per-frame speed profiles were averaged across trials for each participant are shown in Figure 7. We display the relative speed of the ray with respect to the gaze after the angular speed of the gaze drops below 50 deg/s, which, as shown in Figure 7, occurs as the gaze slows down as it reaches the target.

5.2 Analysis of Subjective Responses

After each trial, participants were asked to report how smooth or shaky they found the ray interaction. We refer to these measures jointly as 'selection quality'. After being informed about potential motion artefacts (priming), they were asked to what degree they noticed artefacts, and to what degree they were distracted by those artefacts. We refer to these measures jointly as 'artefact severity'. All responses were collected on a scale from one to seven. The overall distribution of the responses to each of the four questions is shown in Figure 8.

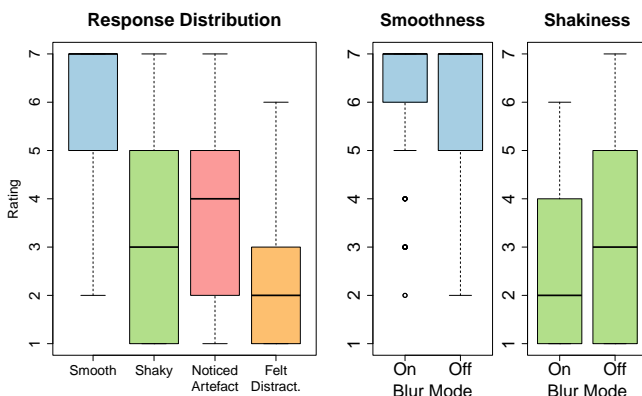


Figure 8: Distribution of participant responses to questions about selection after each trial. The 'Noticed Artefacts' and 'Felt Distracted' characteristics were only queried after priming.

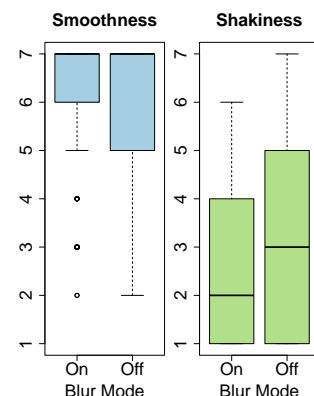


Figure 9: Overall rating distribution for smoothness and shakiness for each blur mode setting. The effect of blur mode on ratings before and after priming is shown in Figures 12 and 13.

Participants' selection performance and questionnaire responses were analysed by performing a repeated-measures ANOVA with blur mode, display refresh rate, target position radius, and priming as independent variables. Since performance data did not satisfy the normality assumption (as indicated by Shapiro-Wilk test) and response data is ordinal, we applied the Aligned Rank Transform [42], which allows multi-factorial analysis of non-parametric data. With the transformed data, we performed the ANOVA for each dependent variable. Where significant effects were indicated, ART-based pairwise comparisons were conducted [12].

5.2.1 Selection Performance

After removing outliers, the mean selection time was 2.29 s ($SD = 0.31$). A significant main effect of target position radius was found on the time that participants took to select the target ($F_{1,450} = 726, p < .001$), with a selection time of 2.10 s ($SD = 0.19$) when $r = 1$ m and 2.48 s ($SD = 0.30$) when $r = 2$ m. No main effects or interaction effects were observed from the other independent variables.

5.2.2 Perceived Selection Quality

A repeated-measures ANOVA on data from all trials indicated significant main effects on reported smoothness and shakiness from blur mode, target radius, and priming. Selection trials with the motion blur effect active were perceived as smoother ($F_{1,450} = 4.90, p < .05$) and less shaky ($F_{1,450} = 9.12, p < .01$), as shown in Figure 9. Target selection during trials with the smaller target position radius $r = 1$ m was also perceived as smoother ($F_{1,450} = 7.52, p < .01$) and less shaky ($F_{1,450} = 22.8, p < .001$) than when $r = 2$ m, as shown in Figure 10. Trials that occurred after priming were perceived as smoother ($F_{1,450} = 23.5, p < .001$) and less shaky ($F_{1,450} = 17.1, p < .001$) as those before priming (Figure 11). No interaction effects were indicated on either characteristic.

The differences in perceived smoothness that were observed before and after priming (Figure 11) could have been caused by order effects, since the primed trials always occurred after the non-primed trials. To quantify the impact of the order effects, we split the dataset into two parts (*before priming* and *after priming*), each containing data from two experiment blocks. We then tested whether the block order had a significant effect on the ratings within each dataset by including the block number in the repeated-measures ANOVA. For ratings of smoothness, no significant effect of block order was found before or after priming. For shakiness ratings, no significant effect of block number was found *after* priming, but a significant effect was found *before* priming ($F_{1,211} = 8.06, p < .01$), indicating that participants gave higher ratings for shakiness during the first experiment block (med = 3) than during the second experiment block (med = 2).

Given the strong differences between ratings in primed and unprimed trials, we also analyse whether the blur mode has a similar effect before and after priming. Before priming, no significant effect of blur mode was found on smoothness, but a significant effect was found on shakiness ratings ($F_{1,211} = 5.66, p < .05$). Pre-priming ratings are visualised in Figure 12. After priming, we find significant effects of blur mode on smoothness ($F_{1,211} = 4.54, p < .05$) and shakiness ($F_{1,211} = 7.17, p < .01$), shown in Figure 13.

5.2.3 Perceived Artefact Severity

No main effects of any of the independent variables were observed on the degree to which artefacts were noticed, and the degree to which participants felt distracted by artefacts.

5.2.4 Blur Mode Preference

To conclude the experiment, participants were asked to indicate which blur mode they preferred. 8 participants preferred the blur-

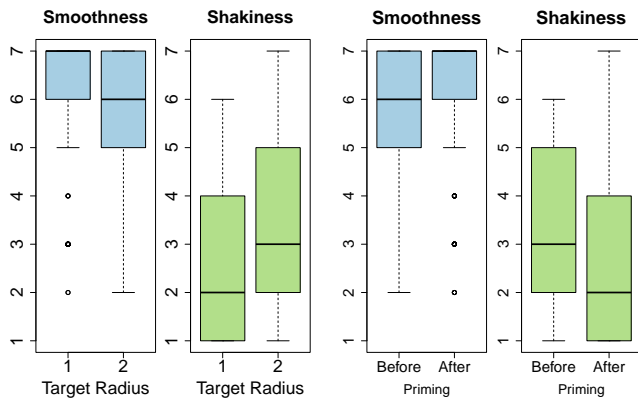


Figure 10: Rating distribution of smoothness and shakiness, depending on the target radius.

Figure 11: Rating distribution for smoothness and shakiness, before and after priming.

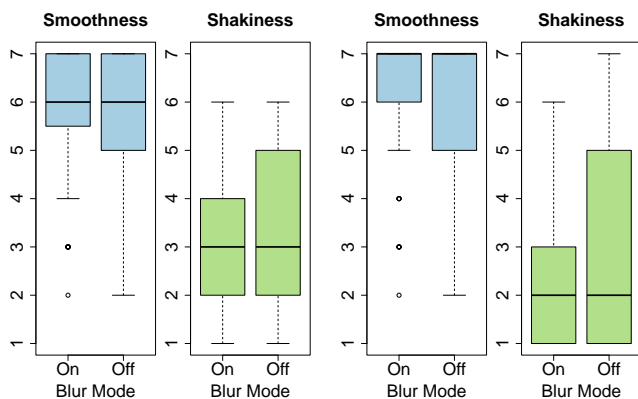


Figure 12: **Pre-priming** effect of blur mode on smoothness and shakiness rating distributions.

Figure 13: **Post-priming** effect of blur mode on smoothness and shakiness rating distributions.

like motion effect, 5 participants preferred the ray with no motion effect, and 2 participants indicated no preference.

5.3 Joint Analysis

Since multiple image artefacts are more severe when stimulus movement is faster [4], we expect faster movement of the ray to result in an increase of reported artefact severity. To test this, we analyse the correlation between the reported motion artefact severity (obtained from participant responses) and the ray movement speed (obtained from tracking data). Ray movement speed was calculated for each selection motion by determining the total angular movement towards the target (see Section 5.1.1) during the time interval between selecting one target and intersecting the next. The speed was then averaged across each trial. Correlation was only investigated for post-priming trials (since artefact severity was not reported before priming) and for trials where target radius $r = 2\text{m}$ (where a longer distance between targets meant that artefacts were more likely to be noticed). Correlation was computed using Spearman's rank correlation coefficient, since reported artefact severity data is ordinal. When no motion blur effect was present, there was a moderate correlation coefficient ($r_s = 0.36$) between ray speed and artefact severity, but this was not significant ($p = .190$, $n = 15$). When the blur mode was present, there was a non-significant correlation coefficient of $r_s = -0.22$ ($p = .426$, $n = 15$).

6 DISCUSSION

In this section, we discuss whether the results presented in Section 5 indicate that motion artefacts occur during a simple ray selection task (Section 6.1), and analyse the impact of the motion blur effect (Section 6.2) and other independent variables (Section 6.3).

6.1 Virtual Ray Pointing and Motion Artefacts

6.1.1 Ray and Gaze Motion

In Section 2.3 we posit that conditions conducive to multiple imaging artefacts are present when a VR user selects objects with a pointing ray. This section discusses the validity of that claim, with reference to the motion paths shown in Figure 6 and the relative speed profiles shown in Figure 7.

In Figures 6 and 7, it is clear that there are significant differences between participants; some direct their gaze and ray to the target almost simultaneously (e.g. P8 and P14), while others reach the target significantly later with the ray than with their gaze (e.g. P1 and P5). It can be observed for many participants that when the gaze has neared the target and slowed down to fixate on the target, the ray is still in motion (e.g. P2 and P4). In Figure 7, the angular speed of the ray towards the target after the saccade is above 30 deg/s for many participants, and much higher in some cases. In a study examining motion artefacts, Mackin et al. [24] found that multiple image artefacts were perceived by more than 50% of participants when a stimulus strobing at 100 Hz moved with a speed of 30 deg/s. As stimulus speed increased above 70 deg/s, more than 95% of participants perceived multiple image artefacts. A lower strobing rate of 60 Hz led to a higher rate of multiple image artefact detection. Given that the relative post-saccade movement of the ray with respect to the gaze that we observed in this study is faster than 30 deg/s for many participants, and the stimuli were presented at a lower frame rate than 100 fps (72 and 90 fps), we believe that the conditions during ray pointing are sufficient to cause multiple image artefacts for many participants that completed the study.

6.1.2 Multiple Image Artefact Detection

Despite the conditions for multiple image artefacts seemingly being present, participants only moderately agreed with the statement "I noticed some motion artefacts", with a median agreement rating of 4 on a scale of 1 to 7. The distribution of responses is shown in Figure 8. The motion blur effect did not significantly change the distribution. The wide distribution could be explained by the varying characteristics of selection movements performed by different individuals. This is hinted at by the analysis in Section 5.3, which shows a (non-significant) trend for a correlation between ray movement speed and motion artefact severity. For some participants, the conditions for multiple image artefacts (i.e. fast relative gaze/ray motion) may not arise, because the ray moves slowly (e.g. P13), or because the ray moves in tandem with the gaze such that little relative motion occurs (e.g. P8). Individual movement characteristics also modulate the *duration* of fast relative gaze/ray motion. If the ray moves fast enough to create artefacts, but only for a fraction of a second, potential artefacts may not be noticed. From the relative speed profiles in Figure 7, it can be seen that fast relative motion occurs for varying periods of time after the saccade ends. We maintain that multiple image motion artefacts are noticeable, at least for some users, although further research (with larger sample sizes) is required to determine the conditions under which motion artefacts are noticed, with particular consideration for individual movement styles. This research would also allow investigation of the effectiveness of artefact mitigation techniques (such as our proposed ray blur method) to be focused on the usage contexts and user groups that are most affected by motion artefacts.

6.2 Motion Blur Effect

The motion blur effect that was evaluated during the user study had the positive effects of increasing the perceived smoothness and decreasing the perceived shakiness reported by participants during a pointing ray selection task. In a direct comparison between blur modes, the ray with the motion blur effect was preferred by a majority of users. No selection performance cost was observed when the blur effect was active, which might have been expected if the additional visual cues that were introduced (i.e. the ray trail) had made it more difficult for participants to judge the current position and trajectory of the ray.

At the same time, the motion blur effect did not result in a significant difference in the degree to which motion artefacts were noticed by participants. The lack of a measurable reduction in reported motion artefacts is surprising and somewhat contradicts the improvement in perceived selection quality (increased smoothness and decreased shakiness). This suggests that the increase in perceived smoothness when the blur effect was active was not caused by mitigation of motion artefacts, but by another desirable quality introduced by the blur effect. It is possible that the blur effect masked the appearance of juddering ray motions that occur due to natural hand movement instabilities or tracking inaccuracies. Some high spatial frequencies caused by the hard edges of the un-blurred ray are removed by the blur effect, making small variations in ray position harder to detect, leading to a smoother visual effect.

Overall, the motion blur effect can be said to have a positive effect on virtual ray selection scenarios, in that it improved the perceived selection quality, particularly when the distance between targets is large. However, our study did not provide evidence that a reduction of motion artefacts as a result of the motion blur effect was the cause of the difference in perceived selection quality.

6.3 Other Independent Variables

The distance between targets increased the selection time, which is expected as the required movement is larger. The negative effect of increasing target position radius on perceived smoothness and shakiness could be indicate that when participants are required to move the ray over longer distances between targets, negative artefacts in the form of non-smooth and shaky motion are more likely to occur and be perceived.

Priming had a positive effect on reported smoothness and shakiness. The post-priming differences could signal a familiarisation effect, since priming took place halfway through the study, by which time participants may have become accustomed to any non-smooth or shaky motions that affected their responses earlier in the study.

Increasing the display refresh rate did not significantly affect the reported detection of artefacts and resulting distraction, nor the perceived selection quality, even though it is expected that higher refresh rates reduce motion artefacts, as shown by Mackin et al. [24]. The lack of significant effect of display refresh rate could be explained by the small difference between the two frame rates used, which in turn was determined by the equipment used for the study.

6.4 Limitations

In this work, we evaluate the effectiveness of a method for applying motion blur to a pointing ray. Our proposed blur method is closely related to previous work on applying motion blur to moving objects [36, 29, 41]. However, real-time motion blur for general scenes is typically implemented as a post-process that utilises screen-space velocity vectors to guide a blurring kernel [27, 16]. The study presented in this work does not compare our blur technique with such methods, and so cannot provide an indication of which blur method is better suited for application to fast-moving pointing rays.

7 CONCLUSION AND FUTURE WORK

This work addresses multiple image artefacts that appear when virtual pointing rays are employed in VR applications. With reference to previous work on temporal aliasing phenomena, we explain why ray pointing actions can produce multiple image artefacts and use tracking data from a user study to show that the necessary conditions for multiple image artefacts arise when users perform a selection task. To mitigate the severity of multiple image artefacts, we develop and implement a motion blur effect suited for approximating motion blur for simple ray visualizations. The effect is tested in a user study where participants undertake a selection task in VR using a ray, and is shown to improve user experience, in that participants reported that selection felt smoother and less shaky when the motion blur effect was active. It therefore serves as a beneficial augmentation in many VR applications that employ pointing rays for selection and navigation.

Further work on pointing rays and motion artefacts should address the link between the duration of artefact visibility and how reliably they are noticed by participants. Investigation of how the allocation of attention to a primary task (e.g. selection) affects participants' perception of motion artefacts is also relevant to quantifying artefact severity.

As significant increases in display refresh rates are unlikely to be prioritised over form factor in the next generations of HMDs, the appearance of motion artefacts and techniques for reducing their severity will remain a relevant issue for developers of VR applications, particularly where fast-moving objects are inherent to the use case. Application areas that could benefit from investigation into the effects of motion blur on motion artefacts are sports and combat games in which balls, projectiles, and weapons can move quickly across the field of view. While users of pointing rays can rely on proprioception to help direct the ray to the target, no equivalent sense is available when judging the trajectory of non-user-controlled objects. Further research could therefore investigate whether application of motion blur effects to moving objects could mitigate motion artefacts and improve perceived motion quality, as well as whether such techniques affect users' judgement of object trajectories.

ACKNOWLEDGMENTS

This work is funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under the project ID 444532506, SPP2236 - AUDICTIVE - Auditory Cognition in Interactive Virtual Environments. We thank the members of the Virtual Reality and Visualization Research Group at Bauhaus-Universität Weimar for their support.

REFERENCES

- [1] F. Argelaguet, C. Andujar, and R. Trueba. Overcoming eye-hand visibility mismatch in 3d pointing selection. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 43–46. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1450579.1450588 1
- [2] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300331 1
- [3] P. Baudisch, E. Cutrell, and G. Robertson. High-density cursor: A visualization technique that helps users keep track of fast-moving mouse cursors. 2003. 3
- [4] P. Bex, G. Edgar, and A. Smith. Multiple images appear when motion energy detection fails. *Journal of Experimental Psychology: Human Perception and Performance*, 21:231–238, 1995. doi: 10.1037//0096-1523.21.2.231 1, 2, 3, 8
- [5] E. Bozgeyikli, A. Raji, S. Katkooi, and R. Dubey. Point & teleport locomotion technique for virtual reality. In A. Cox, P. O. Toups Dugas,

- R. L. Mandryk, and P. Cairns, eds., *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pp. 205–216. ACM, New York, NY, USA, 2016. doi: 10.1145/2967934.2968105 1
- [6] D. Burr. Motion smear. *Nature*, 284(5752):164–165, 1980. doi: 10.1038/284164a0 2
- [7] D. A. Allport. Temporal summation and phenomenal simultaneity: Experiments with the radius display. *Quarterly Journal of Experimental Psychology*, 22(4):686–701, 1970. doi: 10.1080/14640747008401947 2
- [8] S. Daly. *Engineering Observations from Spatiotemporal and Spatiotemporal Visual Models*, pp. 179–200. Springer US, Boston, MA, 2001. doi: 10.1007/978-1-4757-3411-9_9 1
- [9] S. J. Daly, N. Xu, J. Crenshaw, and V. J. Zunjarrao. A psychophysical study exploring judder using fundamental signals and complex imagery. *SMPTE Motion Imaging Journal*, 124(7):62–70, 2015. doi: 10.5594/j18616 1, 5
- [10] J. Davis, Y.-H. Hsieh, and H.-C. Lee. Humans perceive flicker artifacts at 500 hz. *Scientific Reports*, 5:7861, 2015. doi: 10.1038/srep07861 1, 2
- [11] G. Denes, A. Jindal, A. Mikhailiuk, and R. K. Mantiuk. A perceptual model of motion quality for rendering with adaptive refresh-rate and resolution. *ACM Trans. Graph.*, 39(4), aug 2020. doi: 10.1145/3386569.3392411 1
- [12] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock. An aligned rank transform procedure for multifactor contrast tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, p. 754–768. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3472749.3474784 7
- [13] K. Fatahalian, E. Luong, S. Boulos, K. Akeley, W. R. Mark, and P. Hanrahan. Data-parallel rasterization of micropolygons with defocus and motion blur. In *Proceedings of the Conference on High Performance Graphics 2009*, HPG '09, pp. 59–68. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1572769.1572780 3
- [14] J. Gabel, S. Schmidt, O. Ariza, and F. Steinicke. Redirecting rays: Evaluation of assistive raycasting techniques in virtual reality. In *29th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–11. ACM, New York, NY, USA, 2023. doi: 10.1145/3611659.3615716 1, 4
- [15] C. W. Grant. Integrated analytic spatial and temporal anti-aliasing for polyhedra in 4-space. *SIGGRAPH Comput. Graph.*, 19(3):79–84, 1985. doi: 10.1145/325165.325184 3
- [16] J.-P. Guertin, M. McGuire, and D. Nowrouzezahrai. A fast and stable feature-aware motion blur filter. In *Proceedings of High Performance Graphics*, HPG '14, p. 51–60. Eurographics Association, Goslar, DEU, 2014. 3, 9
- [17] W. Hershberger. Saccadic eye movements and the perception of visual direction. *Perception & psychophysics*, 41(1):35–44, 1987. doi: 10.3758/BF03208211 2
- [18] International Organization for Standardization. Ergonomics of human-system interaction: Part 411: Evaluation methods for the design of physical input devices, 2012-05. 4
- [19] P. Johnson, J. Kim, D. M. Hoffman, A. Vargas, and M. S. Banks. Motion artifacts on 240hz oled stereoscopic 3d displays. *SID Symposium Digest of Technical Papers*, 45(1):797–800, 2014. doi: 10.1002/j.2168-0159.2014.tb00209.x 2
- [20] J. Korein and N. Badler. Temporal anti-aliasing in computer generated animation. *SIGGRAPH Comput. Graph.*, 17(3):377–388, 1983. doi: 10.1145/964967.801168 3
- [21] J. Larimer, C. Feng, J. Gille, and V. Cheung. 31:3 judder-induced edge flicker at zero spatial contrast. *SID Symposium Digest of Technical Papers*, 34(1):1042, 2003. doi: 10.1889/1.1832466 1
- [22] J. Larimer, J. Gille, and J. Wong. 41.2: Judder-induced edge flicker in moving objects. *SID Symposium Digest of Technical Papers*, 32(1):1094, 2001. doi: 10.1889/1.1831749 2
- [23] L. Liu, R. van Liere, C. Nieuwenhuizen, and J.-B. Martens. Comparing aimed movements in the real world and in virtual reality. In *Proceedings of the 2009 IEEE Virtual Reality Conference*, VR '09, p. 219–222. IEEE Computer Society, USA, 2009. doi: 10.1109/VR.2009.4811026 2
- [24] A. Mackin, K. C. Noland, and D. R. Bull. The visibility of motion artifacts and their effect on motion quality. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2435–2439. IEEE, Phoenix, AZ, USA, 2016. doi: 10.1109/ICIP.2016.7532796 1, 8, 9
- [25] Mads J.L. Rønnow, Ulf Assarsson, and Marco Fratarcangeli. Fast analytical motion blur with transparency. *Computers & Graphics*, 95:36–46, 2021. doi: 10.1016/j.cag.2021.01.006 3
- [26] S. Mahmoudpour and P. Schelkens. Visual quality analysis of judder effect on head mounted displays. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, A Coruna, Spain, 2019. doi: 10.23919/EUSIPCO.2019.8902665 5
- [27] M. McGuire, P. Hennessy, M. Bukowski, and B. Osman. A reconstruction filter for plausible motion blur. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pp. 135–142. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2159616.2159639 3, 9
- [28] M. R. Mine. Virtual environment interaction techniques, 1995. 1
- [29] Natalya Tatarchuk, Chris Brennan, and John R. Isidoro. Motion blur using geometry and shading distortion. In Wolfgang Engel, ed., *ShaderX2: Shader Programming Tips & Tricks with DirectX 9*. Wordware Publishing, Inc., Plano, Texas, 2004. 3, 4, 9
- [30] F. Navarro, F. J. Serón, and D. Gutierrez. Motion blur rendering: State of the art. *Computer Graphics Forum*, 30(1):3–26, 2011. doi: 10.1111/j.1467-8659.2010.01840.x 3
- [31] A. K. Pääkkönen and M. J. Morgan. Effects of motion on blur discrimination. *JOSA A*, 11(3):992–1002, 1994. 2
- [32] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, 1983. doi: 10.1145/357318.357320 3, 4
- [33] A. Reshetov. Cool patches: A geometric approach to ray/bilinear patch intersections. In E. Haines and T. Akenine-Möller, eds., *Ray Tracing Gems*, pp. 95–109. Apress, Berkeley, CA, 2019. 4
- [34] J. E. Roberts and A. J. Wilkins. Flicker can be perceived during saccades at frequencies in excess of 1 khz. *Lighting Research & Technology*, 45(1):124–132, 2013. doi: 10.1177/1477153512436367 2
- [35] Rongkai Shi, Yushi Wei, Yue Li, Lingyun Yu, and Hai-Ning Liang. Expanding targets in virtual reality environments: A fits' law study, 2023. 4
- [36] J. Schmid, R. W. Sumner, H. Bowles, and M. Gross. Programmable motion effects. *ACM Trans. Graph.*, 29(4), jul 2010. doi: 10.1145/1778765.1778794 3, 9
- [37] T. Scott Murdison, C. McIntosh, J. Hillis, and K. J. MacKenzie. 3-1: Psychophysical evaluation of persistence- and frequency-limited displays for virtual and augmented reality. *SID Symposium Digest of Technical Papers*, 50(1):1–4, 2019. doi: 10.1002/sdtp.12840 5
- [38] K. Sung, A. Pearce, and C. Wang. Spatial-temporal antialiasing. *IEEE transactions on visualization and computer graphics*, 8(2):144–153, 2002. doi: 10.1109/2945.998667 3
- [39] A. B. Watson. High frame rates and human vision: A view through the window of visibility. *SMPTE Motion Imaging Journal*, 122(2):18–32, 2013. doi: 10.5594/j18266XY 2
- [40] S. Wei, D. Bloemers, and A. Rovira. A preliminary study of the eye tracker in the meta quest pro. In *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences*, IMX '23, p. 216–221. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3573381.3596467 6
- [41] M. M. Wloka and R. C. Zeleznik. Interactive real-time motion blur. *The Visual Computer*, 12(6):283–295, 1996. doi: 10.1007/BF01782290 3, 4, 9
- [42] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 143–146. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1978942.1978963 7
- [43] R. S. Woodworth. Accuracy of voluntary movement. *The Psychological Review: Monograph Supplements*, 3(3):i–114, 1899. doi: 10.1037/h0092992 2
- [44] L. Yang, S. Liu, and M. Salvi. A survey of temporal antialiasing techniques. *Computer Graphics Forum*, 39(2):607–621, 2020. doi: 10.

1111/cgf.14018 3

- [45] D. Yu, H.-N. Liang, X. Lu, K. Fan, and B. Ens. Modeling end-point distribution of pointing selection tasks in virtual reality environments. *ACM Transactions on Graphics*, 38(6):1–13, 2019. doi: 10.1145/3355089.3356544 4
- [46] D. J. Zielinski, H. M. Rao, M. A. Sommer, and R. Kopper. Exploring the effects of image persistence in low frame rate virtual environments. In *2015 IEEE Virtual Reality (VR)*, pp. 19–26. IEEE, Arles, France, 2015. doi: 10.1109/VR.2015.7223319 4