Effects of Latency Jitter on Simulator Sickness in a Search Task

Jan-Philipp Stauffert*
University of Würzburg

Florian Niebling†
University of Würzburg

Marc Erich Latoschik‡
University of Würzburg

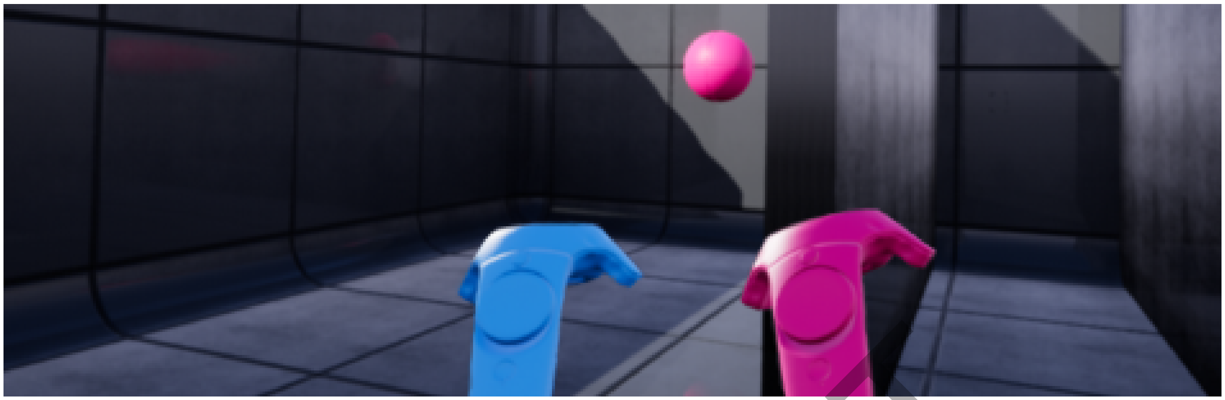

Figure 1: The employed search task. A sphere has to be found that acquires one of two colors if it is in the middle of the field of view, here magenta. Once the sphere acquired a color, the trigger button of the controller with the same color has to be pressed. Afterwards a new sphere has to be found. The task was conducted with and without latency jitter added to the HMD tracking. We found significant differences of the cybersickness with latency jitter present.

ABSTRACT

Low latency is a fundamental requirement for Virtual Reality (VR) systems to reduce the potential risks of cybersickness and to increase effectiveness, efficiency and user experience. In contrast to the effects of uniform latency degradation, the influence of latency jitter on user experience in VR is not well researched, although today's consumer VR systems are vulnerable in this respect. In this work we report on the impact of latency jitter on cybersickness in HMD-based VR environments. Test subjects are given a search task in Virtual Reality, provoking both head rotation and translation. One group experienced artificially added latency jitter in the tracking data of their head-mounted display. The introduced jitter pattern was a replication of a real-world latency behavior extracted and analyzed from an existing example VR-system. The effects of the introduced latency jitter were measured based on self-reports simulator sickness questionnaire (SSQ) and by taking physiological measurements. We found a significant increase in self-reported simulator sickness. We therefore argue that measure and control of latency based on average values taken at a few time intervals is not enough to assure a required timeliness behavior but that latency jitter needs to be considered when designing experiences for Virtual Reality.

Index Terms: D.1.3 [Programming Techniques]: Concurrent Programming—Parallel programming; D.4.8 [Operating Systems]: Performance—Measurements; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 INTRODUCTION

Today's computer systems come with inherent fluctuations of performance. In many applications, especially graphical applications, programmers try to use every bit of performance the computer offers while still expecting that changed data is displayed as fast as possible. Failure to meet these demands results in delayed information which in turn affects user performance and experience. The main measurement to describe the responsitivity of an application is by measuring the end-to-end latency, the time from the user performing an action until this action's results are shown on the display. Between an action and the perception of its effects lie multiple soft- and hardware systems each running at their own pace. Delays introduced anywhere in the system might lead to inconsistencies between action and stimulus presented by the head-mounted display, leading to unpleasant jitter in the VR experience.

There are many factors that influence execution time that are not deterministic or too manifold so they appear non-deterministic. Examples for influencing factors are hardware-based aspects such as interrupts, as well as software-based aspects such as preemption by the operating system or interfering processes or threads. Communication with other devices such as external trackers or other computers in distributed systems is even more prone to fluctuations in execution time.The employment of multi- and manycore systems sacrifices determinism in the order of operations at runtime for increased performance. Collaborating threads and processes constantly compete for resources. While the overall usage of processors may be maximized, in non-hard realtime systems each thread or process is allotted time on a best effort basis. There are no guarantees that a process gets execution time at the time it needs to perform calculations, ultimately leading to missed updates in rendering, or to renderings based on obsolete inputs. Runtime behavior, especially in interactive applications, is therefore difficult to estimate. As well as these influences are included in the mean latency calculation to achieve a preferably stable frame rate, they can still accumulate to create non-deterministic latency spikes.

Most related research describes the effect on task performance and cybersickness of a constant decrease in latency [10]. There are experiments that show decreased task performance in the presence of spatial jitter [23], but few reports on the effects of temporal jitter. While it is shown that periodic jitter patterns provoke sickness [25], the quasi-random behavior of current real-time systems is yet to be

---

*e-mail:jan-philipp.stauffert@uni-wuerzburg.de
†e-mail:florian.niebling@uni-wuerzburg.de
‡e-mail:marc.latoschik@uni-wuerzburg.de

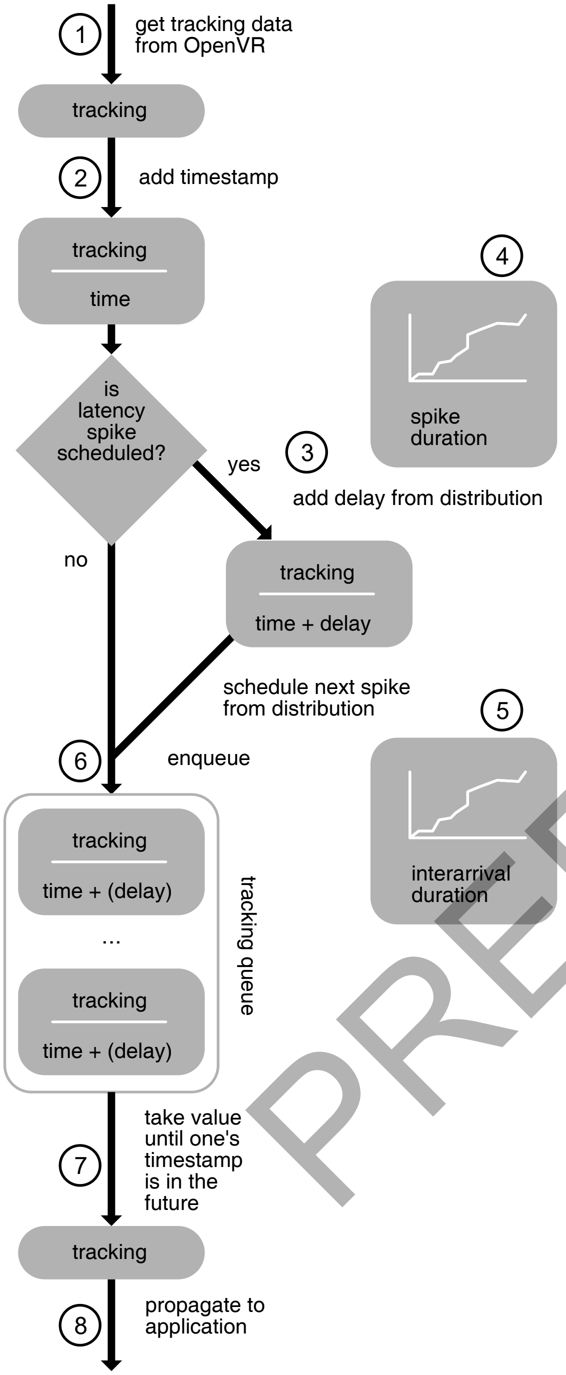

Figure 2: Latency Injection for the experiment. Tracking data is received (1) and annotated with the time of arrival (2). If a latency spike is scheduled, the timestamp is increased (3) by a delay taken from a probability distribution (4). If a spike occurred, a new spike is scheduled whose time is taken from another probability distribution (5). The tracking data is enqueued (6) and immediately dequeued until a timestamp in the future is encountered (7). The tracking data is then forwarded to the application (8).

surveyed. We assume latency jitter to cause similar issues regarding cybersickness.

Our main contributions in this paper are:

- A modification of the Unreal Engine 4 plugin for the HTC Vive to allow the introduction of latency jitter
- A user study evaluating the effect of latency jitter on cybersickness in tasks requiring both continued head movement as well as rotation, finding a significant influence of latency jitter on cybersickness

2 RELATED WORK

Cybersickness is a problem of VR applications where users are experiencing symptoms such as nausea [11]. While some users are more sensitive, there are certain factors that make cybersickness worse for most users. Visual delay was found as a major contributing factor already in early simulators [6]. Latency also influences the performance of test subjects both if time variant latency is added [10] and if latency spikes occur [17, 23]. The assumption is consequently that latency spikes influence cybersickness with a similar impact as the better researched time invariant latency.

Latency has been injected into virtual environments to be able to repeatedly evaluate effects on task performance, presence, and other factors, in controlled experiments. Typically, latency studies delay tracker input data by a controllable amount of time units — frames or multiples of the tracker sampling rate — by employing a ring buffer or other FIFO data structures either inside the tracker itself, its software driver, or the VR application. Experiments are then performed with different, yet most often constant per experiment, amounts of latency artificially injected into the system.

Ellis et. al. tested distinguishability of changes in latency for hand [5] as well as for head [4] movements. They employ custom tracker drivers to ensure a low base latency and to provide the ability to add custom latency to their input devices. Building on this work, Mania et. al. test sensitivity to head tracking latency in virtual environments [14]. Meehan et. al. studied the effects of latency on presence in stressful virtual environments [15]. To enable user studies with different latency settings, they adapted their VRPN client implementation to delay tracker input data by a fixed amount of time to add constant end-to-end latency to their system, enabling controlled experiments with 50ms and 90ms of latency respectively. Other studies that control latency, e. g. performed by Allison et. al. [1], or more recent work on latency control by Papadakis et. al. [16] as well as by Waltemate et. al. [24], also only allow for the insertion of constant latency by delaying tracker input data using ring buffers.

Time invariant latency, however, ignores that latency in applications typically is not constant, but changing over time. This typically happens due to inhomogeneous scene complexity as well as due to different computational complexity of the simulated world in VR given different viewpoints or points in time. Previous work on time varying latency jitter focuses mostly on periodic changes [19, 26]. Periodic patterns are the result of different work cycles in cooperating systems. Here, pattern frequency is shown to play a more important part than amplitude regarding subjective sickness [13]. While periodic latency changes results in time invariant latency, we focus on non-periodic, quasi-random latency behavior. Stauffert et. al. introduced a model for non-periodic latency jitter that allows for injection of latency and latency spikes into VR applications to replicate different latency scenarios [21].

The performance of VR applications is usually assessed by measuring motion-to-photon latency which tracks the time between an input on a certain input channel and the time it takes to show its effect on a display. As with injecting latency, the measurements are used to measure a time-invariant latency. Approaches to measure this latency are sine fitting [22], light sensing [3], and automated frame counting [7].

3 Latency Jitter Injection

We are conducting an experiment to assess the impact of latency jitter on cybersickness. To provide a controlled environment for latency studies, we first describe our method to inject latency jitter into a VR system. Subsequently, we present the user study evaluating task performance under latency scenarios, and discuss our results.

To inject latency jitter, we modify the HTC Vive plugin of the Unreal Engine 4. The built-in plugin is called every frame to update position and orientation of controllers and HMD from tracking data. The plugin queries the OpenVR interface to the device for the current values, making them accessible to the VR application. Instead of directly propagating the received values, we delay these data according to a probability distribution to inject latency jitter. See Figure 2 for a diagram describing the procedure .

After receiving the current tracking data from OpenVR (1), we annotate this acquired sample with the current time (2) . The current time is then compared to the time at which the next latency spike is scheduled to occur (3). If this time is exceeded, a spike duration is drawn from a probability distribution and added to the timestamp (4). The next time a latency spike is to occur is scheduled by drawing a value from the inter-arrival probability distribution and adding it to the current time (5). Independent whether a latency spike has occurred or not, the sample is enqueued into a buffer (6). After the last received value is added to its respective buffer, the buffer is drained until there are no more values in the buffer or a value is encountered whose availability-timestamp is in the future (7). The value taken last gets propagated to the application.

With this mechanism, one value that was received with a scheduled latency spike can block the buffer for the duration of the spike even though there would be samples from the tracker arriving afterwards that are not directly delayed by a scheduled spike.

Latency jitter can have many sources. Among them are unreliable execution times of code due to optimizations such as caching, hyperthreading and multithreading. Additionally there is other software running concurrently, needing CPU time. The plethora of influencing factors makes code execution time unpredictable. While it doesn't exhibit recurring patterns, a model representing jitter distribution and duration is needed for the simulation of latency jitter [21].

To obtain a realistic jitter profile, we base the injected latency on measurements with the VR middleware OSVR which are then scaled to resemble a more pathogenic system. The resulting latency jitter was tested by three experts who found it to realistically resemble under-performing VR systems. The assessment was a subjective feedback based on multiple years of experience with VR systems.

For the base latency distribution, we measured the time that tracking data needs to propagate through the VR middleware OSVR. With a custom device sending and a custom client receiving data, we measure the time in between the sending and receiving.

The result is a characteristic pattern where most values gather around a mean value with only a minuscule population deviating from this mean [20]. Since we only want to observe the effect of latency jitter, we use the outliers for the simulation. There, we restrict the injection to outliers of order two and above, as can be seen in Figure 3, as they represent the 0.1% of measurements that are often ignored in time-invariant latency experiments.

The chosen latency jitter pattern is noticeable for a user who has used the employed VR setup before. The system repeatedly exhibits multiple frame drops in short succession combined with times of lesser impact of latency spikes. The effect is comparable to a computer operating at the edge of its capacity or a VR system whose tracking experiences interference.

Since the scene used in the experiment has no moving parts other than user interaction, it is not discernible whether only the HMD tracking sometimes lags or if the whole VR simulation is halted.

4 Experimental Design

We conduct an experiment to survey the relation of latency jitter to cybersickness using the introduced latency jitter injection method.

The task between groups only varies in the condition if latency jitter is present or not. The elimination of other factors allows us to focus on the connection between latency jitter and cybersickness.

4.1 Hypotheses

As discussed before, latency jitter was regarded in respect to performance in virtual environments but not yet in respect to cybersickness. Cybersickness is a major factor that reduces virtual reality acceptance. Hence, we employ different indicators of cybersickness for our experiment to get a more holistic picture.

For our experiment we pose our hypotheses as follows:

H1: Latency jitter evokes cybersickness

H2: Latency jitter influences physiological measures

The evaluation of H1 is based on the established simulator sickness questionnaire [11]. On a smaller note, we ask participants during the experiment about their well-being. Since the simulator sickness questionnaire is the often employed way to measure cybersickness, it is the major tool to be able to detect if latency jitter can cause cybersickness. The mid experiment questions are less specific but may help to find further evidence of cybersickness.

H2 asks if effects can be measured from body responses. This question is based on research of Kim et. al. [12] and Meehan et. al. [15] who found a significant connection between, among others, heart rate and cybersickness. The experiment in Kim et. al. involved a driving task which is more prone to cybersickness than our method of movement which allows the user to walk freely in the tracked space. Meehan et. al. use the fear of height as a stress inducing factor. While our physiological responses are expected to be smaller, a significant find would justify to establish physiological measures as a further means to determine cybersickness along the simulator sickness questionnaire for latency experiments.

5 Method

5.1 Participants

46 participants were recruited for this study. One had to be excluded due to technical difficulties during the experiment. The final sample consisted of 45 subjects, 36 female, 9 male. The ages ranged from 18 to 31 with a mean of 21.18 and a standard deviation of 2.58. 22 had usable heart-rate data and 28 had usable skin-conductance data. The low yields were a consequence of the constant movement during the experiment impacting the measuring device. All participants gave written informed consent and got course credits for their participation.

5.1.1 Procedure

Participants first filled out a questionnaire containing demographic questions, the motion sickness susceptibility questionnaire [8], the games motivation questionnaire [18] and the simulator sickness questionnaire [11]. They then were equipped with an Empatica E4 wristwatch measuring physiological data (galvanic skin response, heart rate) and the HTC Vive HMD with controllers to proceed with the experiment. The experiment started with five minutes of standing in the virtual scene without movement to get a baseline for the physiological measurements.After the five minute acclimatization phase, the subjects continued with a search task for 9.5 minutes. After three, six and nine minutes of the search task phase, they were asked how much fun they had on a scale from 1 to 5, how immersed they felt on a scale from 1 to 5 based on [2], and how sick they felt on

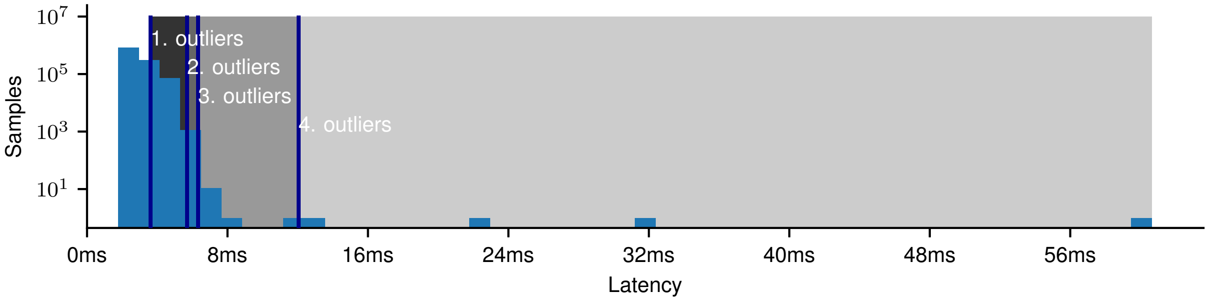

Figure 3: Histogram of the latency durations used to inject latency jitter with outlier orders adapted from [21]. Most latency values(92.72%) gather around a mean with another 7.2695% just above. The remaining 0.0105% contain the outliers that are often ignored.

Outlier Order	Percentage of Samples >=	Min	Max	Mean	Standard deviation
	100.0000%	1.76ms	3.61ms	2.82 ms	0.21ms
1	7.2800%	3.61ms	5.70ms	4.46ms	0.35ms
2	0.0105%	5.70ms	6.22ms	5.84ms	0.14ms
3	0.0020%	6.33ms	7.71ms	6.79ms	0.40ms
4	0.0004%	12.04ms	60.65ms	28.02ms	17.85ms

Table 1: Tabular overview of the jitter data used for the jitter simulation. For the latency injection, we use the data for the outlier orders of two and above which represent less than 0.1% of the measurements.

a scale from 0 to 4. The experiment ended with a one minute phase without movement to again measure physiological data. Afterwards, the simulator sickness questionnaire was filled out a second time.

The five minutes before the search task and the one minute after was chosen to be able to gather physiological data. The data during the search task is unreliable as movement may disturb the measuring process. The exposure time was chosen to resemble Kim et. al.'s procedure [12] for comparability.

5.1.2 Task

The search task was designed as to find spheres appearing in a virtual scene. The users were instructed to focus their viewing direction on the position of a sphere and press a button for the next sphere to appear. The spheres appear at random in one of four corridors which cannot be surveyed completely from a static position. Hence, movement is required to find the next sphere. The movement method was walking in the physical environment as tracked by the HMD position. This is the most natural way to express movement.

The scene was a room with an open ceiling. In front of the user were three walls, forming four corridors - two between the walls and one on each side. Confer Figure 5 for an overview of the setting. Spheres could spawn between the walls, as well as on either side. Only one sphere was present in the scene at the same time. The position was chosen at random for each sphere. The walls obstructed the view provoking a left and right movement to find a sphere.

Once a sphere was found, it had to be in the middle of the user's field of view to acquire one of two possible highlighting colors, magenta or blue. The test if the sphere was looked at was performed by an intersection test with a cone originating in the users HMD, forcing the users to focus on the sphere.

If the looked-at sphere was highlighted in either magenta or blue by user focus, the user needed to press the trigger button on the respective controller. For this, the left and right controllers were colored in blue and magenta, respectively. The controller's colors stayed the same for the whole experiment duration.

If the user pressed the trigger button of the controller with the same color as the sphere, a success message was presented. If they used the other controller's trigger button, a failure message was presented. Either way, a new sphere appeared somewhere else in the scene. The amount of right and wrong detections as well as a timer was shown above the walls.

The task as well as the feedback was chosen to suggest a competitive environment and to distract from the real intention of provoking head rotation and translation. The random position of each sphere forced a constant movement from left to right and back, spanning the whole three meters of tracked area. Users were instructed to not move forward or backward, to keep their overview of the scene comparable to that of the other participants.

Users were divided into two groups of participants. One group experienced added latency jitter on the HMD tracking, while the other did not. The latency jitter was only added in the 9.5 minutes of the search task. Both the acclimatization phase as well as the minute afterwards was free from intervention. Due to the test subjects not moving with a reduced amount of head movement in the absence of the search task, they were supposed to not detect the different conditions between the phases.

5.2 Apparatus

The experiment was conducted on a computer with an Intel i7-6700K processor and a NVidia GTX 1080.

The physiological measurements were done with an Empatica E4 wristwatch. The watch communicated with the computer via Bluetooth. The data was provided to the VR application by the Empatica Bluetooth server.

Using frame counting, we determine our system's base motion to photon latency to be 35.67ms. For this the controller and the monitor were filmed with a high speed camera at 240Hz. In the resulting video, the frame difference between the movement of the real controller and the virtual controller was counted to receive a motion to photon latency between the controller and the monitor. Multiple measurements were taken with the mean of 46.67ms. This number was adjusted by the difference in reactivity of the HMD

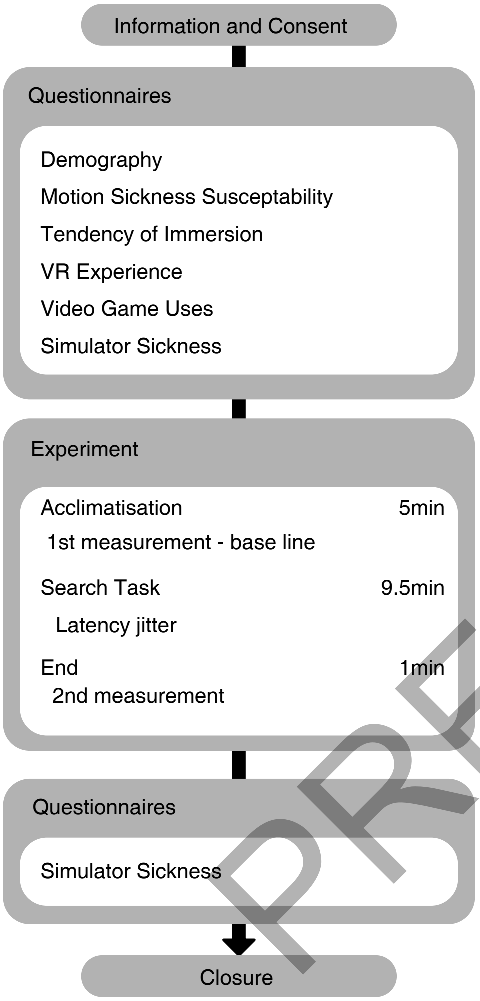

Figure 4: Illustration of the experiment procedure. Participants filled out the simulator sickness questionnaire before and after the experiment. Before and after the experiment, there is a time of standing in VR without performing any tasks, to gather physiological measurements.

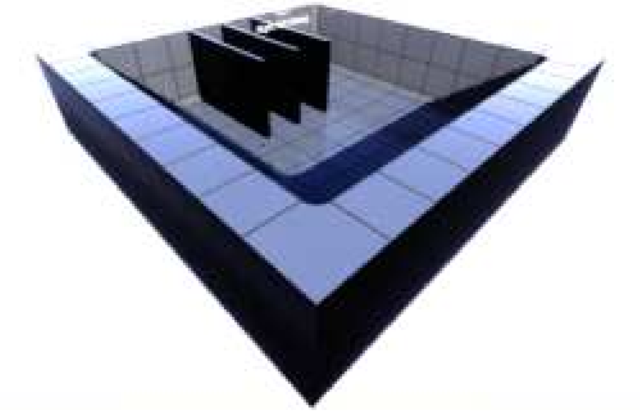

Figure 5: Illustration of the virtual scene. One sphere appears at a time in between the walls. The walls block the sight to force the user to move left and right to spot the spheres. A counter with time and successful trials distracts from the purpose of the study by suggesting a competitive environment. The walkable area is smaller than the virtual room to avoid walking through walls. The colors are adjusted for better visibility.

screen to the monitor screen again gathered with frame counting. There, colors were shown on both the monitor and the HMD and recorded with a high speed camera at 1000Hz. The frame difference between the monitor and the HMD resulted in 11.0ms, resulting in the aforementioned 35.67ms delay from motion to HMD photon latency. The detour via the monitor was necessary as it is hard and error prone to track the movement of the virtual controller on the HMD screen with a camera. The difference in frames per second of the camera video between the two measurements was chosen because for the first take, the video needed spatial resolution to determine the movement. The second video needed only to compare the time between colors shown on the displays, requiring less spatial resolution and allowing the camera to switch to a mode with increased temporal resolution.

6 LIMITATIONS

Even though we base our jitter profile on actual measurements, it is yet unclear if the scaling of values creates realistic conditions. Consulted experts report the experience concerning responsiveness to being close to experiences with unmodified, uncontrolled VR systems under heavy load.

Using the Empatica E4 wristwatch allows to gather physiological measurements. However, it can't be as accurate as arrays of electrodes used in other experiments. The gathering of measurements is influenced by movement. We tried to limit movement in the phases before and after the experiment but were not able to eliminate them.

7 RESULTS

Close to half of the test subjects ($n = 10$) in the latency jitter condition didn't notice the repeated lag in tracking of the HMD. Those who noticed reported it to be very obvious and annoying.

Each scale was analyzed by separately applying a mixed-design analysis of variance (split-plot ANOVA) with the between factor *latency jitter condition*. Generalized $\eta^2(\eta_g^2)$ is reported as a measure of effect size.

Four subjects were taken out of the analysis for their high up front sickness score. The simulator sickness questionnaire total score shows significant results for the comparison of both groups ($F_{1,39} = 4.44$, $p < 0.041$, $\eta_g^2 = 0.102$). The symptom clusters nausea ($F_{1,39} = 0.79$, $p < 0.38$, $\eta_g^2 = 0.020$) and oculomotor ($F_{1,39} = 1.02$,

		Pre		Post		Delta	
		Mean	SD	Mean	SD	Mean	SD
SSQ	NL	9.69	6.68	7.30	6.95	-2.38	6.10
	L	7.17	4.56	9.33	7.68	2.16	7.59
SSQ O	NL	25.01	15.18	17.43	14.15	-7.58	13.24
	L	19.49	10.33	16.96	16.93	-2.53	18.31
SSQ D	NL	16.70	16.03	13.22	15.95	-3.48	16.82
	L	11.27	12.15	20.55	17.94	9.28	18.33
SSQ N	NL	18.13	18.55	13.83	17.91	-4.29	12.94
	L	14.99	13.35	14.99	15.84	0.00	17.59

Table 2: Values for the SSQ and subscales before and after the experiment. `NL` and `L` are for the condition without latency added and with latency jitter added.

$p < 0.32$, $\eta_g^2 = 0.025$) did not test significant while the disorientation cluster ($F_{1,39} = 5.38$, $p < 0.026$, $\eta_g^2 = 0.121$) did.

Mid experiment questions didn't test significant for fun ($F_{1,39} = 0.21$, $p < 0.21$, $\eta_g^2 = 0.001$), immersion ($F_{1,39} = 0.90$, $p < 0.39$, $\eta_g^2 = 0.005$) and sickness ($F_{1,39} = 0.00$, $p < 0.99$, $\eta_g^2 < 0.0001$).

We analyzed the physiological data analogous to Meehan et. al. [15]. They take the delta between the mean value of the baseline and the mean value of the stressful environment. For our analysis, we discard the first minute of the acclimatization phase and the first minute of the experiment. The omission of the two minutes is to reduce the effect of the transition from the real world to the virtual world and from the relaxed phase to the active search task phase in the measurement data. The mean values are compared with a t-test between the groups with Cohen's d as an indicator of effect size. The skin-conductance values are drift corrected with a regression line derived from the base line in the first five minutes.

With the four subjects taken out due to high sickness scores in the beginning, there are 22 usable measures for heart rate (11 without latency added, 11 with latency jitter) and 28 usable measures for skin conductance (11 without latency added, 17 with latency jitter). Heart-rate did test significant ($p < .037$, $d = 0.95$) while skin-conductance ($p < 0.39$, $d = 0.34$) did not.

8 DISCUSSION

With the significant result of the simulator sickness questionnaire comparing pre- and post-conditions between the groups with and without latency jitter, we accept hypothesis H1. Looking at the sub-scales, we find latency jitter affecting disorientation, while there is no significant result on the nausea and oculomotor sub-scales. The mid-experiment questions how sick the participants feel did not find a difference between conditions. As this question points in nausea direction, it affirms the not found significance of the simulator sickness questionnaire nausea sub-scale.

The significant finding in the simulator sickness total score forms the basis to show that latency jitter is a problem that cannot be overlooked. In addition to the base latency that is known to have implications, repeated spikes in system latency causes uneasiness even though they are only of short duration.

We tried to avoid techniques known to provoke more cybersickness such as certain travel techniques (driving, joystick movement) or other-directed camera movement. Other than the base cybersickness that VR is known to produce, we tried to have latency jitter as the only source of cybersickness. We assume that latency jitter can interact with other cybersickness inducing causes, though we leave this as future research.

We did find a significant correlation between heart rate and cybersickness. This is in line with the research of Kim et. al. [12] and Meehan et. al. [15]. Skin conductance, however wasn't conclusive. The result has to be evaluated under the small sample size. The constant movement during the experiment rendered a lot of measurement unusable. A lot of subjects didn't have any usable physiological data. The usable heart rate measurements for subjects included in the analysis were few (mean = 97.05, $\sigma = 57.51$ during the search-task).

It is notable how many test subjects didn't notice the repeated delays in head tracking. Pre-studies with colleagues showed that the introduced latency jitter was obvious. We assume that experience with virtual reality applications plays an important role. Although most test subjects already experienced VR before, many did so only once or twice with an exposure time of roughly an hour. Among those, some experienced systems that had a worse performance than our experiment in the latency jitter condition. They presumably see HMD-based VR systems as inherently affected by latency jitter. It is unclear if an adaptation to jitter happened, even though the random nature of the latency jitter is expected to prevent adaption. The simplicity of the task may have helped to adapt better.

Our sample contains more women than men. Graeber et. al. [9] finds that there is no gender difference with respect to cybersickness as other research has assumed. The susceptibility of cybersickness differs between individuals. Therefore, we cannot argue how much the employed latency jitter affects a person but only that there is an effect.

Qualitative discussions with test subjects after the experiment as well as the discussion above suggests that latency jitter is experienced in different ways depending on prior VR experience. Depending on the success of virtual reality as a technology to reach a wider user base, latency jitter needs to be reevaluated with users more accustomed to VR.

The task between groups only varied in the condition if latency jitter is present or not. The observation of this is that there is a connection between cybersickness and latency jitter. It will be interesting to see how latency jitter compares and interacts with other cybersickness inducing factors especially time invariant latency in future work.

9 CONCLUSION

This paper investigated the effect of latency jitter on cybersickness. We developed a latency extraction and injection system which allows to measure time-variant latency patterns in existing VR applications and to later inject these patterns in arbitrary target applications. This system was used to develop and execute a novel experiment to evaluate the effect of latency jitter on cybersickness in HMD-based VR. To our best knowledge, this is the first approach to assess the effect of latency jitter in VR-systems.

The implementation of the latency jitter injector for the Unreal Engine 4 allows to perform further experiments with different patterns of latency jitter and to also modulate overall latency values. The described method is easy to adapt for alternative implementations and target systems and provides a general method for further research on latency jitter and its effects.

Conducting a user study with a search task, we found a significant difference between the group with and without latency jitter added. The task was designed to provoke constant head rotations and translations. We conclude that not only time invariant latency, but also latency spikes cause unwanted implications, provoking cybersickness.

With the experimental results documenting a significant impact of latency jitter on cybersickness, we argue that more research needs to be done to understand the implications of latency jitter. Better ways to measure latency jitter need to be found, which can be used to inject latency jitter in more controlled environments to understand the effects and to possibly develop countermeasures. This research is a first step demonstrating the applicability of the developed tools to enable further research on the effects of latency jitter in VR.

REFERENCES

[1] R. S. Allison, L. R. Harris, M. Jenkin, U. Jasiobedzka, and J. E. Zacher. Tolerance of temporal delay in virtual environments. In *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, VR '01, pp. 247–. IEEE Computer Society, Washington, DC, USA, 2001.

[2] S. Bouchard, J. St-Jacques, G. Robillard, and P. Renaud. Anxiety increases the feeling of presence in virtual reality. *Presence: Teleoperators and Virtual Environments*, 17(4):376–391, 2008.

[3] M. Di Luca. New method to measure end-to-end delay of virtual reality. *Presence*, 19(6):569–584, 2010.

[4] S. R. Ellis, M. J. Young, B. D. Adelstein, and S. M. Ehrlich. Discrimination of changes in latency during head movement. In *Proceedings of the HCI International '99 (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Communication, Cooperation, and Application Design-Volume 2 - Volume 2*, pp. 1129–1133. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1999.

[5] S. R. Ellis, M. J. Young, B. D. Adelstein, and S. M. Ehrlich. Discrimination of changes of latency during voluntary hand movement of virtual objects. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 43, pp. 1182–1186. SAGE Publications Sage CA: Los Angeles, CA, 1999.

[6] L. H. Frank, J. G. Casali, and W. W. Wierwille. Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 30(2):201–217, 1988.

[7] S. Friston and A. Steed. Measuring latency in virtual environments. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):616–625, 2014.

[8] J. F. Golding. Motion sickness susceptibility questionnaire revised and its relationship to other forms of sickness. *Brain research bulletin*, 47(5):507–516, 1998.

[9] D. A. Graeber and K. M. Stanney. Gender differences in visually induced motion sickness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, pp. 2109–2113. SAGE Publications Sage CA: Los Angeles, CA, 2002.

[10] Z. Ivkovic, I. Stavness, C. Gutwin, and S. Sutcliffe. Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3d Shooter Games. pp. 135–144. ACM Press, 2015. doi: 10.1145/2702123.2702432

[11] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[12] Y. Y. Kim, H. J. Kim, E. N. Kim, H. D. Ko, and H. T. Kim. Characteristic changes in the physiological components of cybersickness. *Psychophysiology*, 0(0):050826083901001–???, Aug. 2005. doi: 10.1111/j.1469-8986.2005.00349.x

[13] A. Kinsella, R. Mattfeld, E. Muth, and A. Hoover. Frequency, not amplitude, of latency affects subjective sickness in a head-mounted display. *Aerospace medicine and human performance*, 87(7):604–609, 2016.

[14] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill. Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity. In *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, APGV '04, pp. 39–47. ACM, New York, NY, USA, 2004. doi: 10.1145/1012551.1012559

[15] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks. Effect of latency on presence in stressful virtual environments. In *IEEE Virtual Reality, 2003. Proceedings.*, pp. 141–148, March 2003. doi: 10.1109/VR.2003.1191132

[16] G. Papadakis, K. Mania, and E. Koutroulis. A system to measure, control and minimize end-to-end head tracking latency in immersive simulations. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp. 581–584. ACM, 2011.

[17] K. S. Park and R. V. Kenyon. Effects of network characteristics on human performance in a collaborative virtual environment. In *Virtual Reality, 1999. Proceedings., IEEE*, pp. 104–111. IEEE, 1999.

[18] J. L. Sherry, K. Lucas, B. S. Greenberg, and K. Lachlan. Video game uses and gratifications as predictors of use and game preference. *Playing video games: Motives, responses, and consequences*, 24(1):213–224, 2006.

[19] M. E. St. Pierre, S. Banerjee, A. W. Hoover, and E. R. Muth. The effects of 0.2hz varying latency with 20100ms varying amplitude on simulator sickness in a helmet mounted display. *Displays*, 36:1–8, Jan. 2015. doi: 10.1016/j.displa.2014.10.005

[20] J.-P. Stauffert, F. Niebling, and M. E. Latoschik. Reducing application-stage latencies of interprocess communication techniques for real-time interactive systems. In *Virtual Reality (VR), 2016 IEEE*, pp. 287–288. IEEE, 2016.

[21] J.-P. Stauffert, F. Niebling, and M. E. Latoschik. Towards comparable evaluation methods and measures for timing behavior of virtual reality systems. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pp. 47–50. ACM, New York, NY, USA, 2016. doi: 10.1145/2993369.2993402

[22] A. Steed. A Simple Method for Estimating the Latency of Interactive, Real-time Graphics Simulations. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 123–129. ACM, New York, NY, USA, 2008. doi: 10.1145/1450579.1450606

[23] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and S. I. MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pp. 43–50. IEEE, 2009.

[24] T. Waltemate, I. Senna, F. Hülsmann, M. Rohde, S. Kopp, M. Ernst, and M. Botsch. The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pp. 27–35. ACM, 2016.

[25] M. L. Wilson. *The effect of varying latency in a Head-Mounted Display on task performance and motion sickness*. PhD thesis, Clemson University, 2016.

[26] W. Wu, Y. Dong, and A. Hoover. Measuring Digital System Latency from Sensing to Actuation at Continuous 1-ms Resolution. *Presence: Teleoperators and Virtual Environments*, 22(1):20–35, Feb. 2013. doi: 10.1162/PRES_a_00131