Measuring System Visual Latency through Cognitive Latency on Video See-Through AR devices

Robert Gruen[1], Eyal Ofek[1], Anthony Steed[1,2], Ran Gal[1], Mike Sinclair[1], and Mar Gonzalez-Franco [*1]

[1]Microsoft Research, USA

[2]University College London, UK

Figure 1: Latency was tested both through a hardware instrumentation-based measurement (bottom left) and the new cognitive latency technique (bottom right) on four different devices. The first device, Prism, was an ad-hoc system that attached a pair of colour cameras to an Acer Windows Mixed Reality device (1 top left). This system aimed at providing a top end video see-through quality. We also tested Oculus Quest (2), Oculus Rift S (3) and the Valve Index (4). On the Bottom Left, hardware instrumentation-based measurement setup. Cameras (A and B) are synchronized to capture at the exact same time by the board (2). While (1) is a clock running at sub-millisecond accuracy. The clock for camera (B) is seen by the HMD (3), which displays the video see-through. The scene was kept well illuminated to reduce automatic exposure time problems of the HMD cameras. On the Bottom Right, a participant performing the rapid decision making task while wearing a see-through VR headset.

ABSTRACT

Measuring Visual Latency in VR and AR devices has become increasingly complicated as many of the components will influence others in multiple loops and ultimately affect the human cognitive and sensory perception. In this paper we present a new method based on the idea that the performance of humans on a rapid motor task will remain constant, and that any added delay will correspond to the system latency. We ask users to perform a task inside different video see-through devices and also in front of a computer. We also calculate the latency of the systems using a hardware instrumentation-based measurement technique for bench-marking. Results show that this new form of latency measurement through human cognitive performance can be reliable and comparable to hardware instrumentation-based measurement. Our method is adaptable to many forms of user interaction. It is particularly suitable for

*Corresponding author: margon@microsoft.com

systems, such as AR and VR, where externalizing signals is difficult, or where it is important to measure latency while the system is in use by a user.

Index Terms: Human-centered computing—Virtual reality—;—— Computing methodologies—Perception—

1 INTRODUCTION

System latency can be detrimental to the experience of many interactive setups from touch displays [2, 5] through to virtual reality (VR) systems [13, 18]. The term "latency" refers to a lag somewhere in a system. It is determined by measuring the time difference at two different locations at which a signal goes through the "system" components [12]. The system latency can then be considered to be the accumulation of all the latencies across components that go from the the generation of a signal until it reaches the human. The measurements of these internal electronic signals can be measured using photonic and electronic hardware instrumentation-based measurement techniques.

The time for systems to respond to user actions is a complicated combination of transport and synchronization delays of the multiple

components inside the system. For VR, those can include latencies from tracking devices [18], visual rendering [13], haptic rendering [34] and audio [24].

Latency affects subjective experience and presence of VR scenarios as well as task performance during physical interactions or collaborative tasks [28]. Significant delays can produce simulator sickness [11]. The effects of latency are therefore quite interlinked to the human perceptual system [20, 23].

Indeed, humans have an intrinsic latency in their sensory systems [6, 7, 25]. Typical sensory-motor responses depend on the input modality. Visual stimuli are first integrated at a neuronal level after approximately 200 ms from stimulus [22]. Auditory signals can be integrated within 100 ms of stimulus [47]. Additional delay in cognition is added in the motor response itself. This creates response times closer to 300 ms in visual Eriksen flanker rapid response tasks [14, 43]. These afferent paths are embedded in the motor control models [19, 20]. When body semantic violations have been described, it is generally assumed that the full re-afferent loop till motor action is integrated at 400 ms [43, 45]. In a task where the reaction of the user is requested we would normally expect to have a minimal response time between 300 and 400 ms. Response times over this level could be expected to be a result of other system delays that are not due to human processing of the incoming signals.

Interestingly, cognitive processing remains quite uniform over repetitions of a task and thus and can be averaged over a number of trials. Of course, different people will have different latency as they will have a different expertise and reaction time. However, if the task is tested within subjects across multiple devices, the theory of human perception tells us we should be able to assume the processing will be constant and that any difference in latency will be due to the differential latency between systems.

With this in mind we design a perceptual-motor task and experiment that will test whether cognitive latency could have enough consistency to actually measure visual latency of augmented reality (AR) systems. To validate our theory, we compare the latency obtained during the task against a more traditional numeric (electronic clock-based) approach.

We use video see-through AR devices for multiple reasons: (i) they provide access for good hardware instrumentation-based measurements to have robust system comparisons; (ii) they enable an AR experience while enjoying the field of view of VR [9] in which we can blend with a real world view [26, 30, 56] and modify its content if required [51].

1.1 Contributions

The main contributions of this paper are:

1. We present a new strategy for determining system latency by measuring Cognitive Latency. This method can be a way to measure total system latency when it is hard to instrument the input and measure the corresponding output signals. In this case we are employing video see-through AR.

2. We provide a standardized canonical test to perform cognitive latency tests. From the aspect of accuracy, this inferred method might suffer relative to a direct measurement but it could measure multiple latencies: not only the photon-to-photon latency, but also the tracking latency. In this paper we focus on the photon visual latency. However, the method presented could work with other senses just as well - there is nothing inherently visual in the method - a user is exposed to a stimulus via the system and reports perceiving it.

2 RELATED WORK

2.1 Direct Impacts of Latency

Latency can have a significant impact on the user experience inside VR and AR systems. One of the most obvious impacts is the dis-played images may lag behind head movements. In a VR this may lead to an effect sometimes referred to as 'swimming' [1], and in an AR it can lead to virtual and real objects moving relative to each other, or 'slipping' [42].

Human interaction is also delayed due to added system latency. For example, reaching or pointing tasks are delayed in the video input to the human. This can be investigated from a information throughput point of view based on the classic Fitts's law of human movement in human-computer interfaces [16]. Mackenzie and Ware modelled the effect of latency on difficulty of reaching tasks and suggested that it had a multiplicative effect on difficulty [37]. This was then extended to VR displays where a similar model held for motion in all three directions [54]. Jitter in latency can also have a detrimental impact on similar tasks [53].

Given that system latency has an impact on the user experience, significant engineering effort has been devoted to reducing or hiding latency. Users also appear to be sensitive to differences in latencies of approximately 15ms [38]. However, unconscious effects appear to occur at lower levels. On 2D tasks, Kadowaki et al. found that there was no difference in performance in a direct touch task at 4.3 and 24.4ms [32]. Friston et al. found that performance was perhaps worse, or at least not better, at very low latencies, possibly explained by behaviour of the human motor system having inherent latency [17]. While this might mean that for VR, it is not clear if latencies need to be lower than currently supported on modern consumer systems (e.g. less than 20ms is suggested as a desirable target for VR and the Oculus systems [15], for AR, especially optical see-through, to maintain registration between virtual and real objects, latencies will need to be lower than 1ms (e.g. see [10]). This level has been validated by other studies on touch pointing tasks which have found that users can identify 1ms latency and 10ms latency starts to affect their performance [31].

2.2 Latency Measurement

The measurement of latency in immersive systems, and more generally in any type of computer system, has long been a concern of developers. A common measurement strategy is to use an external reference which can monitor physical movement or change. Liang et al. measured the latency of a Polhemus Isotrak magnetic tracking system by videoing the tracker on a pendulum together with a screen showing a timestamp of the latest received tracking information. By reconstructing the pendulum motion from the tracking and noting the lowest point on the pendulum swing in the video, the latency could be approximated [36]. Mine provided an overview of sources of latency in a system and a technique for measuring latency that has been highly influential on later work: using a first photodiode to detect a tracked pendulum swinging past its lowest point and a second photodiode to detect the lowest point on the rendered version of the tracked motion [41]. An oscilloscope was used to measure time differences. A similar approach was used by Papadakis et al. where a rotary encoder signal is compared to a photodiode placed on a screen [44]. These approaches involve extra equipment and some constraints on the movements of the device. Di Luca removed the need for constrained motion by having a photodiode attached to the tracked object that moved in front of a image with a grayscale gradient [13]. By showing a virtual world where the gradient also moved depending on the tracking readings, and having photodiode read this display, this produced two varying signal levels which differ in phase. This does require a rather specific display to be created, and thus it is difficult to use in a complete system where rendering time might be an important part of the latency.

One can simply use a video camera to observe both real and virtual motion and note, for example, turning points in the motions [29]. This is somewhat sensitive to the accuracy of identification of matching points, and can only give latency in multiples of the frame time. A slightly different approach is to match a periodic motion between

the real and virtual recordings. Swindells et al. proposed videoing a tracked turntable along with a virtual turntable [52]. The angle between the two turntables times the angular speed gives the latency if the rotation speed is constant. Steed proposed a simpler method that used a video of a pendulum, but mathematically fitted a sine wave to both real and virtual motion, thus extracting latency from the difference in phase [49]. Getting a single video that observes both the real and virtual scene can be a challenge, especially for a head-mounted display (HMD). With modern high-speed cameras, this type of outside observer camera-approach can be made real-time and with resolution beyond the frame rate. (e.g. 1kHz) so that it can report near-continuous latency values [55].

A key point in latency measurement is being precise about the measurement method as different methods can return different results [18]. It is also worth noting that latency can vary over time, for example due to varying frame rates due to differing rendering loads when the user looks in different directions. It is important to sample over multiple events.

Specifically for video-see through AR systems, there are different components causing various delays, in particular the video pass through and the virtual graphics. The latency of the former might be lower [46] or higher [50], depending on the relative speeds of image capture and image rendering. Thus one of the two images might be delayed so as to synchronize real and virtual images.

An important specification in a video see-through AR system is the latency at which the real world video is transmitted to the display. This is very similar to glass-to-glass measurements for video transmission. For example Bachhuber et al. present a simple system on an Arduino micro computer comprising an LED and a photodiode that can detect the LED's image on the screen [3]. To extend this to a system that can detect the latency of every frame of view, one strategy is to encode the time in the video image. Sielhorst et al. use a pattern of moving dots [48], whereas Billeter et al. suggest a pattern of dots [8]. Our benchmark system measurement approach is similar to these last two approaches, but to simplify, we built a 10 kHz LED clock showing elapsed time in milliseconds.

3 MATERIALS AND METHODS

3.1 Devices

We tested four different HMDs with video see-through modes, see Figure 1. The first device was an ad-hoc system, refereed to as Prism, that attached a pair of colour cameras to an Acer Windows Mixed Reality device. We also tested an Oculus Quest which is an untethered VR system and the Oculus Rift S and the Valve Index which are both tethered VR HMDs (Figure 2).
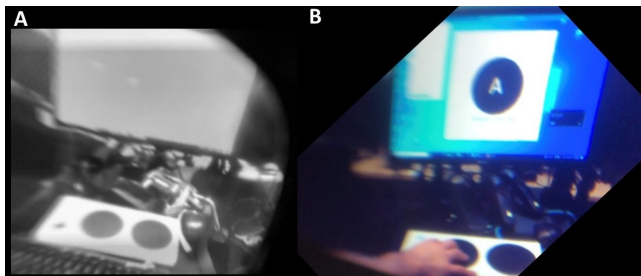

Figure 2: Image as seen through the display of an Oculus Quest (A) and the Prism setup (B). Valve Index provided similar video quality and color than the Prism setup (B). While Oculus Rift S was also black and white like (A).

For the HMDs that required Unity to interface the camera and the display, we created executables to reduce any latency introduced at run-time. However, this needs to be taken into account, as using Unity might increase the latency. Nevertheless, we believe that

using an interfacing program such as Unity would be a normal use to access the cameras and produce synthetic video for future rendering of see-through and mixed reality systems that combine both video and synthetic content [26, 51].

3.2 Oculus Specs

Both the Oculus Quest and Oculus Rift have dual displays at resolutions of 1440x1600 per eye at 72Hz (Quest) and LCD 1280x1440 per eye at 80Hz (Rift).

Both the Oculus Quest and Oculus Rift S HMDs use low resolution gray level cameras (5 in the rift and 4 in the quest) for tracking the position of the HMDs and motion controller accessories. Some or all of these camera feeds can be used by both devices to display the "real-world" as a backdrop, allowing the user to be aware of the physical environment.

These tracking cameras are rarely positioned near the user's eyes pointing forward, which leads us to believe that the backdrop images are the result of some depth-dependent re-projection and correction of original pixels. We do not have exact information regarding as to if and how this projection is performed, but the fact that these images are calculated on the device and fed directly to the display, without the need to upload them into a host machine, leads us to assume very low-latency. The specifications of the cameras are unavailable online to our knowledge.

3.3 Valve Index Specs

The Valve Index comes with dual 1440x1600 LCDs displays running at 90Hz and 2 960x960 global shutter RGB (Bayer) Cameras running at 60Hz. The camera feeds can be accessed on the host machine as a colour video that can be redirected to the HMD using Unity.

3.4 Prism Specs

This device was an ad-hoc system that attached a pair of colour cameras to an Acer Windows Mixed Reality device. The Acer MR device has dual LCD panels with a display resolution of 1440x1440 per eye at 90 Hz.

This device was designed to create a close to optimal see-through AR experience, with low latency and minimising the amount of re-projection between cameras and the display. The connection between the cameras and the display was facilitated via Unity. The cameras were calibrated and connected to the screens in a similar way to previous see-through displays [26, 51].

The two cameras used on the Prism device were custom built using Omni Vision 4689 sensors [40]. The sensors were configured to capture video at 90 Hz at a resolution of 1704 x 1440 pixels. These cameras were connected to the host PC via high-speed USB ports. The camera sync input was wired directly to a Vsync signal generated on the display board of the Acer HMD whose displays run at the same frequency.

3.5 Cognitive Latency

We use a rapid response task similar to the Eriksen flanker task [14] but with a focus on reducing errors rather than inducing them [43].

Participants (n=16, mean age=44, sd=8.5) pressed a large physical button in front of them while they viewed a rendered circle on the computer screen (Figure 1). At a random time after they pressed the button (between 500ms to 3s), the circle changed from white to black and the participants were instructed to remove their hand from the button. They repeated this task until they had completed 10 error-free trials for each HMD. An error was noted whenever users released a button too early. There was a total error rate of 2%. Errors were not counted as trials, so after errors each participant had accomplished at least 10 trials without errors. The data was also cleaned of outliers whenever there was a response time higher than 2 times the standard deviation of the performance per participant and device. The number of removed outliers was 29, which represents

5% of the trials. These numbers together show that participants were quite attentive and waiting for the signal.

This task is performed both directly at the computer screen, as well via a see-through VR HMD. The PC used black and white rendering for the stimuli for all tests so as to be compatible with devices such as the Oculus Rift S and Oculus Quest that use monochrome cameras. See Figure 2.

The system latency can be calculated for each device assuming the baseline recorded against a computer (see Figure 7). That baseline would be the minimal cognitive latency measured during the task. To measure this baseline we asked participants to perform the task directly in front of the computer screen (without wearing any HMD) and we measured the time passed since we rendered the circle till the trigger of the button signal. The same time measurement was used when HMDs where used as the interface to visualize the PC screen. It is true however that the PC could have some minimal lag, on average below the millisecond.

3.6 Numerical Clock Method

We measure the latency of the system with a hardware instrumentation-based measurement method based on a sub-millisecond accuracy clock. For this experiment we ran the clock display at 10kHz; this showed millisecond digits and tenths of milliseconds in the format *NNN.N*.

Using the clock we measure the system latency through a photon-to-photon metric by comparing the captured time at a particular instant from a video frame captured from a camera observing the eyepiece video screen inside the AR system and also directly observing the 10kHz clock. In general, any difference between the clock numbers through the two cameras, the one pointing through the HMD and the one observing the clock would represent the photo-to-photon latency of the complete system. See measurement setup in Figure 1.

In order to synchronize the two cameras to shoot images at the exact time we built a second board and validated the accuracy by firing the cameras without any HMD using the setup in Figure 3.
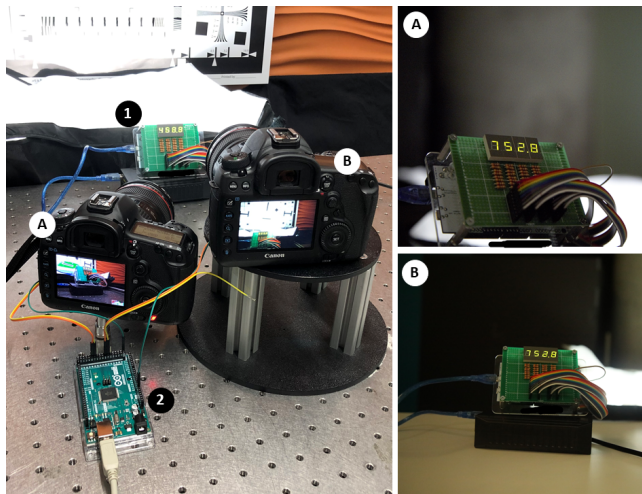

Figure 3: The cameras were observed to be in synchrony with each other to within a half a millisecond. In the image we show how we validate the synchronization: A and B are synchronized to capture at the exact same time by the board 2. On the right the images captured a particular time.

The clock timer, driven by an Arduino Due board running at 84MHz (Figure 4), is capable of updating the clock LED display at in excess of 100kHz. The Due supports 54 digital IO pins, 32 of which were directly wired to four individual seven-segment LEDs. These

digits represented: hundreds, tens, singles, and tenths of elapsed clock pulses. The clock starts from zero and continues to display elapsed time since the device was powered on. The LED turn-on and turn-off time is sub-microsecond [35], therefore we know that our clock is sufficient for at least 0.1 millisecond measurements as the LEDs turn on and off faster than we are updating the digit segments.
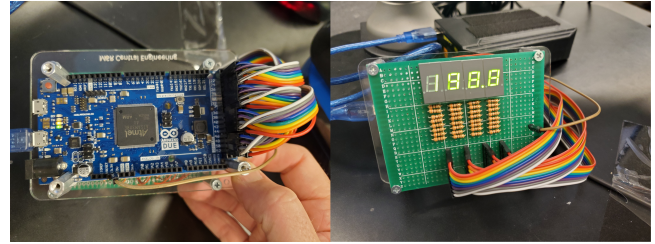

Figure 4: Millisecond accurate clock based on Arduino Due.

The camera shutter is synchronized via an external Arduino Mega 2560 board controller. This camera controller is able to use the remote shutter command port on each camera to trigger the shutter to expose both external cameras at the same time. It is possible to synchronize different camera models/manufacturers by having the controller introduce artificial delays between shutter triggers such that both cameras expose simultaneously. Our setup used two Canon 5D Mark IV with a 16-35mm F4 lens. Both cameras were set to manual focus and had shutter speeds set to 1/4000 of a second.

Once everything has been placed in the necessary configuration, the last remaining step is to take multiple, randomly timed pictures of the clock running (Figure 6). Multiple captures will also help us ensure the precision of the measurement for cases when the display and the camera where not in sync.

The computation of the latency is then straight forward. From the wall clock time (t_1) subtract the time indicated on the headset display time (t_2). The remainder should be the photo-to-photon latency of the system (L).

$$L = t_1 - t_2 \qquad (1)$$

3.7 Synchronization of HMD and Camera Capture

The last outstanding item to consider is the synchronization of the camera capture and the HMD display signal. If the two are not synchronized there is drift between when the camera acquisition is made and the Vsync signal of the display device. This will manifest itself by varying multiple blank measurements where the image of the clock is not captured on the camera through the HMD. There are two possible solutions: 1) synchronize the camera and display signals, 2) take multiple randomly timed images using the method described here and use the lowest observed value.

In order to have enough images to quantify the variance in our image captures, we left the capturing system on for approximately 30 minutes for each device being photographed. During this period, the camera controller was set to record a frame every 5.5 seconds which resulted in approximately 325 measurements. Of those measurements, only a subset were legible due to the aforementioned camera-display sync disparity.

3.8 HMD See-Through Exposure Times

A final consideration: some see-through cameras have auto-exposure times which makes it very difficult to capture latency measurements precisely. Generally, we have found that the observed exposure time on these camera modules to be quite high, above 10ms, determined by the ambient light present. As such, when you attempt to measure latency, the clock timer becomes over exposed and it becomes difficult to read the clock digits thereby introducing less precision in the

measurement (see Figure 5). In these cases you can take multiple measurements and attempt to estimate the approximate value and take their average to give you a rough latency value. However, we found the best way to ensure the exposures are short was to raise the ambient light level. See the added light behind the clock in Figure 3.
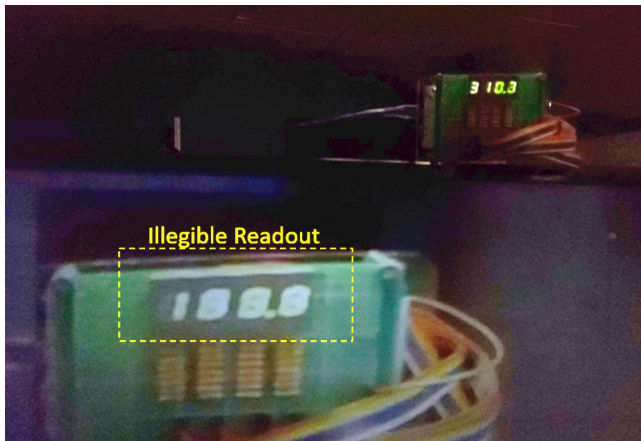

Figure 5: Example of attempted measurement using AR camera that has "long" exposure versus the wall clock resolution. Notice the inability to read the tens and singles millisecond digits.

3.9 Alternative Hardware Instrumentation-Based Measurement Method

We propose an alternate method for capturing latency measurements whenever having two cameras is not possible. Instead of having a second camera, it may be possible to have a second display instead. We can inject a signal splitter (HDMI/Displayport) on the headset tether so that we could view the headset display signal on an external display at the same time than the clock so it is possible to make a measurement with a single camera (Figure 6).

Figure 6: Setup for the alternate hardware instrumentation-based measurement method. The screen shows a mirror of the actual HMD display at 120Hz. While the real Arduino Clock is standing right in front of the cameras. This image is taken directly from the measurement camera.

This approach may introduce additional latency into the system as the splitter device may have some overhead in splitting the signal. Additionally, displays also usually have some frame buffer which

could result on a larger variability of the signal and added noise. This approach should be used sparingly unless the latency of the splitter and display are known.

If the aforementioned splitter approach is not viable we could capture the display from the headset as shown on the host PC via a preview window. The preview window may also introduce additional latency because the host system may be not be presenting the same content to the preview window as is being sent to the headset. As such, this method is the least precise of the three proposed here.

Other items to consider are the refresh rate, response time, and display technology (global shutter, rolling shutter, etc.) of the displays being used. They may differ in different HMDs. Most LCD display panels have a gray-to-gray response time between 4ms and 5ms. For our alternate approach verification, we utilized a 120Hz OLED monitor with a turn-on and turn-off time of less than 1ms. This means our resulting measurements are accurate to within the millisecond range.

In order to avoid multiple exposures of the digits on the display it is necessary to limit the exposure time of the recording camera. For our purposes we used an exposure setting of 1/4000th of a second which results in a exposure time of 0.25ms.

This method could be of use when the two camera system is not a possibility. However it would require more measurements to achieve a similarly accurate latency metric.

4 RESULTS

4.1 Cognitive Latency Task

All participants completed the same rapid reaction task while wearing VR HMDs. We measured the delay in their response from the different devices. This raw cognitive latency can be seen in Figure 7.
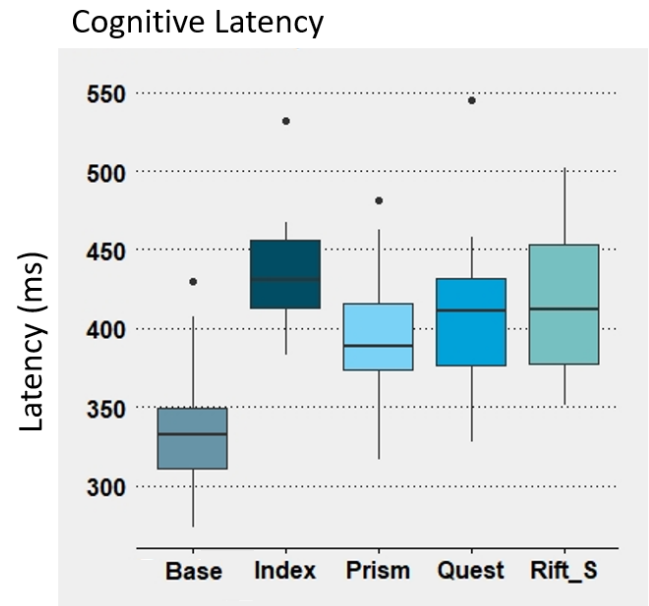

Figure 7: The cognitive latency measured during the task.

Overall, when performing the action directly on the PC, participants responded within 335 milliseconds (sd=11). This is aligned with prior research that has found response times within the range of 300 ms, as reported in visual Eriksen flanker rapid response tasks [14, 43].

The task was completed in the different devices in a counter balanced order, to avoid learning effects as well as concentration or fatigue.

On the other devices, participants exhibited reaction times well over 350 milliseconds. However, the real interest of this task was not to calculate just the reaction times on the different devices but the actual system latency. Our hypothesis then is that the cognitive latency will remain constant for a particular task for a given participant and that any added latency to the reaction time would be the expression of the relevant system latencies.

With these reaction times we were able our infer the system latency which was basically the reaction time on a particular system minus the reaction time on the PC. The system latencies calculated through this method are shown in Figure 8 and Table 1.
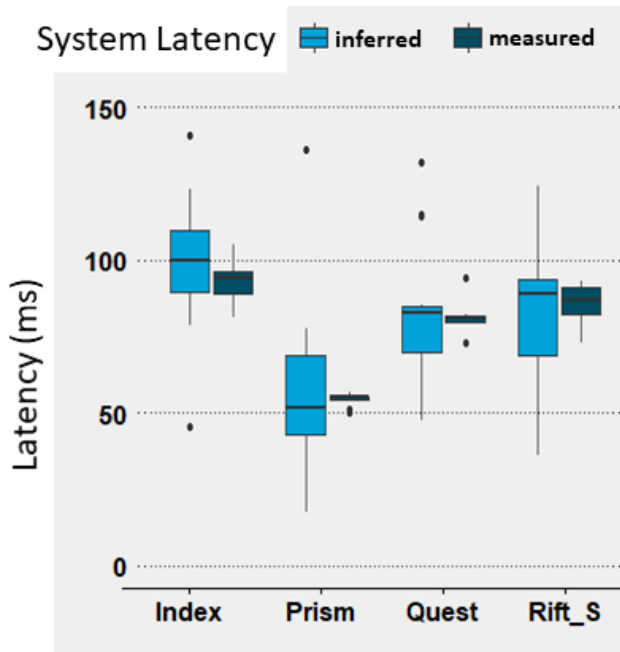

Figure 8: The system latency calculated through both the cognitive inferred method and through the hardware instrumentation-based measurement system for each device.

It is possible that through this calculation the actual latency of the HMDs are a bit faster, since the PC display can have some latency (of at least 4 ms). However as the standard error is over 5ms for all of the devices, this would hide any PC display latency.

4.2 Numerical Latency Measurement

We measured the photon to photon latency using the clock system described on the material section. We captured images of the clock every 5.5 seconds during 30 minutes for each device. Due to the refresh rate of the HMDs, only a fraction of all the pictures taken showed the clock through the see-through video. In total the resulting measurement phase resulted in 6 valid measurements for the Oculus Quest, 9 for the Oculus Rift, 22 for the Prism, and 17 for the Index headset. The Prism headset had a much higher rate of valid pictures due to it's AR cameras and displays being synced to the Vsync of the displays.

Having multiple measurements allowed also us to detect the empirical precision of the hardware instrumentation-based measurement method. Overall, we expect this method to have small deviations as the cameras are synchronized to within 0.5 milliseconds. The measured results can be found in Table 1 and Figure 8.

---

[1] PC (Baseline) response time: 335 ± 11

Table 1: Latency across all the devices and methods.

Latency	Device ($ms \pm SE$)			
	Index	Prism	Quest	Rift S
Cognitive [1]	434 ± 9	394 ± 10	411 ± 12	422 ± 12
System (Inferred)	98 ± 5	58 ± 6	75 ± 8	87 ± 7
System (Measured)	94 ± 2.1	54 ± 1.9	81 ± 2.5	85 ± 2.1

4.2.1 Are the methods comparable?

We analyze whether the system latency calculated through the cognitive task is comparable to the hardware instrumentation-based measurement findings.

We ran Welch Two sample t-tests between all the results and find no significant differences between the two methods. Index: t= 0.8, df=19.1, p=0.42, CI 95% [-7.2, 16.7]. Prism: t= 0.6, df=15.1, p=0.55, CI 95% [-10, 18.11]. Quest: t= -0.65, df=17.7, p=0.52, CI 95% [-25, 13.2]. Rift S: t= 0.19, df=17.5, p=0.84, CI 95% [-15.1, 18.1].

Therefore we assume the calculations of system latency through both methods were comparable. Variability between the two methods is still the biggest differentiator, but we believe the larger variability in the data seen in the cognitive task inferred method will decrease as more participants are added.

4.2.2 Which device has less latency?

We are then interested in analyzing whether the observed differences in latency between devices was statistically significant to find out which system exhibited less latency.

The Repeated Measures ANOVA was run on the latencies with a 4 level factor (device) and grouped within subjects. There was a significant difference between the devices $F(3, 56) = 5.5, p = 0.002$. A post-hoc pairwise comparisons t-test showed that the Index was significantly worse in latency than the Quest (p=.01) but not the Rift S (p=.18). There were no significant differences neither between the Quest and the Rift S (p=.22). Meanwhile our ad-hoc see-through system, Prism, was the top performer being significantly better than all the consumer devices (p less than .02).

Similar results were found when analyzing the hardware instrumentation-based measurement method results. Welch tests showed that the Index implementation of the see-through AR was significantly worse in latency than the Quest (p=.01). Using the hardware instrumentation-based measurement method, the Index implementation was also significantly worse than the Rift S (p=.01). There were no significant differences between the Quest and the Rift S (p=.29). Meanwhile our ad-hoc see-through system, Prism, was the top performer significantly better than all the consumer devices (p less than .03).

A Bartlett test ($K^2 = 3.4, p = 0.3$) showed homogeneity of variances so the data can be analysed with parametric measures (t-tests and RMANOVA).

We believe there are multiple reasons for the difference between the Oculus HMDs and the Valve Index. To start, the Oculus devices do not feed back the video to the PC as it is rendered directly from the signal inside the HMD. On the contrary, the Index allows for the video feed access through the computer and it has to be rendered by adding a layer in Unity. This would add significant latency.

5 DISCUSSION

We looked at two different ways to measure a video AR system's visual latency. On one hand, we can use accurate sensors to measure the exact timing of a certain visual stimulus and compare it to another sensor attached to the HMD display. On the other hand we can

compare the reaction time of users to a visual stimulus with and without wearing the HMD.

The methods have different strengths and weaknesses. An accurate synchronized measurement of the original signal and the display of the HMD is not trivial. There is a need to place a sensor inside the HMD (most typically without a user wearing it) to capture the signal trough the HMD optical system, and at the same time have a highly synchronized sensor sensing the visual stimulus (which is more reliable than using any internal PC event as a trigger). The differences between the displays in size and distance makes a use of a high frame rate camera a challenge.

On the other hand, using users to estimate the latency does not require any additional hardware. The user's reactions may be used to deduce when they see the stimulus in their HMDs displays or on the PC screen. Note that Oculus Quest does not have a separate screen display. However, Oculus have recently released an Oculus Link cable that connects the Quest to a PC that generates the graphics for display. Note that the latency issues here will be very different because the Quest will act as a screen and sensor rig.

The use of a rapid task means that a wide variety of games could potentially include a latency measurement task. Users would not have to be at the lab to report latency, but could be playing in their living room. Reaction time to stimuli could be used to generate a sampling from a large audience while using a particular application. There may be new opportunities for further optimization of software, or to remote diagnose the state of a VR system. However, large numbers of users might be needed to evaluate the system latency if this is variable. The variability within a participant could also be confused by variable latency on the system and this needs to be taken into account (Figure 8). However, through larger number of participants or samples one could potentially reduce the noise of data gathered from individual differences [21] and then use the rest of the variability as a dynamic range.

An important aspect to look at in future work is user sensitivity to external distractions. Potentially HMDs might help reduce the total latency of reaction of a user. This could be due to their limited field of view that may limits the motions of their eyes and reduction of distractions around the user, in particular around the motion sensitive peripheral vision. Our experiment was handled in a lab, minimizing such effects, but verifying this potentially interesting and useful side effect of using HMDs is part of a future work.

5.1 Cognitive task

The rapid reaction task here presented was optimized for visual latency detection. It could be considered a canonical task that could be extended to incorporate other components of VR, such as tracking latency. For example the task might require the person to move around before seeing the release signal.

Additionally, the task could also be adapted for VR systems and existing VR content. Indeed it is possible to envisage running tests on consumer equipment that has been released already. For example, are players of beat saber [4] faster with one particular VR configuration?

A limitation of the use of cognitive tasks is precisely this dependency on the use of multiple people when comparing latency. It seems that inter-individual differences should be taken into account [21]. However it is not clear if it could be more practical to use a single person with more trials across all devices (similar to our work here) than multiple people with fewer samples to average the between persons variations who don't try all the devices.

Furthermore, if in the future system latency was to drop below the millisecond scale, this type of cognitive latency measurement could no longer be possible. On the other hand, that would mean that there is no longer perceivable lag that can affect the users.

Another possible method to measure cognitive latency without a motor task, might be based on attentional responses. For example, exploiting the P300 measurement, which is a positive voltage happening in the central part of the scalp at 300 milliseconds after a stimuli [25, 27]. This would be particularly useful for games that do not require immediate measurable user reactions. As eye gaze is being added to HMDs (e.g. HoloLens 2, Vive Eye Pro), this might also be used to measure attentional responses [33, 39].

Finally, instead of using video stimuli, it may be interesting to use audio. Audio could be displayed through the VR system into the HMD, or it could be mixed into the audio signal bypassing the VR system. The difference in reactions times could then be assessed while the user is using the application and it does not require removing the HMD. This has advantages over our current protocol which needs the user to do the baseline in front of a screen.

6 CONCLUSION

We presented a new method to measure latency based on the idea that the performance of humans on a rapid motor task will remain constant, and that any added delay will correspond to the system latency.

Using cognitive tasks enables measurement of latency in an ecological manner, using actual applications while the users are wearing the HMDs. We believe that this suggests a new class of latency technique where controlled interaction in 3D user interfaces could be re-purposed to measure latency. Furthermore we suggest that there is an opportunity to crowd-source latency data across many users of a particular game or experience.

Such a technique may enable the measuring the latency in different contexts, as well as measuring effects on users. More experiments will be needed to determine if we can utilize different cognitive tasks such as more complex motions, interactions or navigation.

In future work, we hope to extend this method to measure in-situ latency on other types of mixed-reality system. In particular we are interested in extending the method to pure VR setups. This will necessarily involve a more complex visuo-motor task as it will be important to factor tracking latency as part of the user experience, whereas our current work has focused only on the visual latency of a particular class of AR system.

REFERENCES

[1] R. Allison, L. Harris, M. Jenkin, U. Jasiobedzka, and J. Zacher. Tolerance of temporal delay in virtual environments. In *Proceedings IEEE Virtual Reality 2001*, pp. 247–254. IEEE Comput. Soc, Yokohama, Japan, 2001. doi: 10.1109/VR.2001.913793

[2] G. Anderson, R. Doherty, and S. Ganapathy. User perception of touch screen latency. In *International Conference of Design, User Experience, and Usability*, pp. 195–202. Springer, 2011.

[3] C. Bachhuber and E. Steinbach. A system for high precision glass-to-glass delay measurements in video communication. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2132–2136, Sept. 2016. ISSN: 2381-8549. doi: 10.1109/ICIP.2016.7532735

[4] Beat Games. Beat Saber. http://beatsaber.com/, 2019.

[5] F. Bérard and R. Blanch. Two touch system latency estimators: high accuracy and low overhead. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, pp. 241–250, 2013.

[6] C. C. Berger and M. Gonzalez-Franco. Expanding the sense of touch outside the body. In *Proceedings of the 15th ACM Symposium on Applied Perception*, p. 10. ACM, 2018.

[7] C. C. Berger, M. Gonzalez-Franco, E. Ofek, and K. Hinckley. The uncanny valley of haptics. *Science Robotics*, 3(17):eaar7010, 2018.

[8] M. Billeter, G. Röthlin, J. Wezel, D. Iwai, and A. Grundhöfer. A LED-Based IR/RGB End-to-End Latency Measurement Device. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 184–188, Sept. 2016. doi: 10.1109/ISMAR-Adjunct.2016.0072

[9] M. Billinghurst, A. Clark, G. Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[10] A. Blate, M. Whitton, M. Singh, G. Welch, A. State, T. Whitted, and H. Fuchs. Implementation and Evaluation of a 50 kHz, 28μs Motion-to-Pose Latency Head Tracking Instrument. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1970–1980, May 2019. doi: 10.1109/TVCG.2019.2899233

[11] T. J. Buker, D. A. Vincenzi, and J. E. Deaton. The effect of apparent latency on simulator sickness while using a see-through helmet-mounted display: Reducing apparent latency with predictive compensation. *Human factors*, 54(2):235–249, 2012.

[12] G. Casiez, S. Conversy, M. Falce, S. Huot, and N. Roussel. Looking through the eye of the mouse: A simple method for measuring end-to-end latency using an optical mouse. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 629–636, 2015.

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

[14] B. A. Eriksen and C. W. Eriksen. Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & psychophysics*, 16(1):143–149, 1974.

[15] Facebook Technologies LLC. VR Best Practices, 2019. Available at: https://developer.oculus.com/design/latest/concepts/bp-rendering [Accessed November 14, 2019].

[16] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954. doi: 10.1037/h0055392

[17] S. Friston, P. Karlström, and A. Steed. The Effects of Low Latency on Pointing and Steering Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 22(5):1605–1615, May 2016. doi: 10.1109/TVCG .2015.2446467

[18] S. Friston and A. Steed. Measuring latency in virtual environments. *IEEE transactions on visualization and computer graphics*, 20(4):616–625, 2014.

[19] S. Gallagher. Philosophical conceptions of the self: implications for cognitive science. *Trends in cognitive sciences*, 4(1):14–21, 2000.

[20] M. González Franco. Neurophysiological signatures of the body representation in the brain using immersive virtual reality. 2014.

[21] M. Gonzalez-Franco, P. Abtahi, and A. Steed. Individual differences in embodied distance estimation in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 941–943. IEEE, 2019.

[22] M. Gonzalez-Franco, A. I. Bellido, K. J. Blom, M. Slater, and A. Rodriguez-Fornells. The neurological traces of look-alike avatars. *Frontiers in human neuroscience*, 10:392, 2016.

[23] M. Gonzalez-Franco and J. Lanier. Model of illusions and virtual reality. *Frontiers in psychology*, 8:1125, 2017.

[24] M. Gonzalez-Franco, A. Maselli, D. Florencio, N. Smolyanskiy, and Z. Zhang. Concurrent talking in immersive virtual reality: on the dominance of visual speech cues. *Scientific reports*, 7(1):3817, 2017.

[25] M. González-Franco, T. C. Peck, A. Rodríguez-Fornells, and M. Slater. A threat to a virtual hand elicits motor cortex activation. *Experimental brain research*, 232(3):875–887, 2014.

[26] M. Gonzalez-Franco, R. Pizarro, J. Cermeron, K. Li, J. Thorn, W. Hutabarat, A. Tiwari, and P. Bermell-Garcia. Immersive mixed reality for manufacturing training. *Frontiers in Robotics and AI*, 4:3, 2017.

[27] H. M. Gray, N. Ambady, W. T. Lowenthal, and P. Deldin. P300 as an index of attention to self-relevant stimuli. *Journal of experimental social psychology*, 40(2):216–224, 2004.

[28] C. Gunn, M. Hutchins, and M. Adcock. Combating latency in haptic collaborative virtual environments. *Presence: Teleoperators & Virtual Environments*, 14(3):313–328, 2005.

[29] D. He, D. H. Fuhu, D. Pape, G. Dawe, and D. S. Video-Based Measurement of System Latency. In *International Immersive Projection Technology Workshop*, 2000.

[30] Y. Itoh, J. Orlosky, M. Huber, K. Kiyokawa, and G. Klinker. Ost rift: Temporally consistent augmented reality with a consumer optical

see-through head-mounted display. In *2016 IEEE Virtual Reality (VR)*, pp. 189–190. IEEE, 2016.

[31] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the sigchi conference on human factors in computing systems*, pp. 2291–2300. ACM, 2013.

[32] T. Kadowaki, M. Maruyama, T. Hayakawa, N. Matsuzawa, K. Iwasaki, and M. Ishikawa. Effects of low video latency between visual information and physical sensation in immersive environments. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST '18*, pp. 1–2. ACM Press, Tokyo, Japan, 2018. doi: 10.1145/3281505.3281609

[33] S. Kishore, M. González-Franco, C. Hintemüller, C. Kapeller, C. Guger, M. Slater, and K. J. Blom. Comparison of ssvep bci and eye tracking for controlling a humanoid robot in a social environment. *Presence: Teleoperators and virtual environments*, 23(3):242–252, 2014.

[34] B. Knorlein, M. Di Luca, and M. Harders. Influence of visual and haptic delays on stiffness perception in augmented reality. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 49–52. IEEE, 2009.

[35] T. P. Lee. Effect of junction capacitance on the rise time of led's and on the turn-on delay of injection lasers. *The Bell System Technical Journal*, 54(1):53–68, Jan 1975. doi: 10.1002/j.1538-7305.1975.tb02825.x

[36] J. Liang, C. Shaw, and M. Green. On Temporal-spatial Realism in the Virtual Reality Environment. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology*, UIST '91, pp. 19–25. ACM, New York, NY, USA, 1991. event-place: Hilton Head, South Carolina, USA. doi: 10.1145/120782.120784

[37] I. S. MacKenzie and C. Ware. Lag As a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pp. 488–493. ACM, New York, NY, USA, 1993. event-place: Amsterdam, The Netherlands. doi: 10.1145/169059.169431

[38] 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. event-place: Los Angeles, California, USA. doi: 10.1145/1012551.1012559

[39] S. Marwecki, A. D. Wilson, E. Ofek, M. Gonzalez Franco, and C. Holz. Mise-unseen: Using eye tracking to hide virtual reality scene changes in plain sight. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 777–789. ACM, 2019.

[40] Microsoft Research. Project Prism, 2019. Available at: https://www.microsoft.com/en-us/research/project/prism/ [Accessed November 19, 2019].

[41] M. R. Mine. Characterization of End-to-End Delays in Head-Mounted Display Systems. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1993.

[42] M. Nabiyouni, S. Scerbo, D. A. Bowman, and T. Höllerer. Relative Effects of Real-world and Virtual-World Latency on an Augmented Reality Training Task: An AR Simulation Experiment. *Frontiers in ICT*, 3, 2017. doi: 10.3389/fict.2016.00034

[43] G. Padrao, M. Gonzalez-Franco, M. V. Sanchez-Vives, M. Slater, and A. Rodriguez-Fornells. Violating body movement semantics: Neural signatures of self-generated and external-generated errors. *Neuroimage*, 124:147–156, 2016.

[44] 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*, VRCAI '11, pp. 581–584. ACM, New York, NY, USA, 2011. event-place: Hong Kong, China. doi: 10.1145/2087756.2087869

[45] E. F. Pavone, G. Tieri, G. Rizza, E. Tidoni, L. Grisoni, and S. M. Aglioti. Embodying others in immersive virtual reality: electro-cortical signatures of monitoring the errors in the actions of an avatar seen from a first-person perspective. *Journal of Neuroscience*, 36(2):268–279, 2016.

[46] J. P. Rolland and H. Fuchs. Optical Versus Video See-Through Head-Mounted Displays in Medical Visualization. *Presence*, 9(3):287–309,

June 2000. doi: 10.1162/105474600566808

[47] B. A. Rowland, S. Quessy, T. R. Stanford, and B. E. Stein. Multisensory integration shortens physiological response latencies, 2007.

[48] T. Sielhorst, W. Sa, A. Khamene, F. Sauer, and N. Navab. Measurement of absolute latency for video see through augmented reality. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 215–220, Nov. 2007. doi: 10.1109/ISMAR.2007.4538850

[49] 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. event-place: Bordeaux, France. doi: 10.1145/1450579.1450606

[50] A. Steed, Y. W. Adipradana, and S. Friston. The AR-Rift 2 prototype. In *2017 IEEE Virtual Reality (VR)*, pp. 231–232, Mar. 2017. ISSN: 2375-5334. doi: 10.1109/VR.2017.7892261

[51] W. Steptoe, S. Julier, and A. Steed. Presence and discernability in conventional and non-photorealistic immersive augmented reality. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 213–218. IEEE, 2014.

[52] C. Swindells, J. C. Dill, and K. S. Booth. System Lag Tests for Augmented and Virtual Environments. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pp. 161–170. ACM, New York, NY, USA, 2000. event-place: San Diego, California, USA. doi: 10.1145/354401.354444

[53] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *2009 IEEE Symposium on 3D User Interfaces*, pp. 43–50, Mar. 2009. doi: 10.1109/3DUI.2009.4811204

[54] C. Ware and R. Balakrishnan. Reaching for Objects in VR Displays: Lag and Frame Rate. *ACM Trans. Comput.-Hum. Interact.*, 1(4):331–356, Dec. 1994. doi: 10.1145/198425.198426

[55] 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

[56] J. J. Yang, C. Holz, E. Ofek, and A. D. Wilson. Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 1093–1107. ACM, 2019.