Tampere University

Inka Juntunen

SEE-THROUGH LATENCY IN MIXED REALITY HEADSETS

Bachelor's Thesis
Faculty of Information Technology and Communication Sciences
December 2022

ABSTRACT

Inka Juntunen: See-Through Latency in Mixed Reality Headsets
Bachelor's Thesis
Tampere University
Electrical Engineering
December 2022

Extended reality and especially mixed reality is an emerging technology, with a wide range of possible useful applications. A few of these are entertainment and infotainment, education and medical applications. Because there is a wide range of possible beneficial applications, the parameters affecting the usability of the devices want to be measured as accurately as possible.

One thing that has a high effect on the usability of a mixed reality headset is the latency of the device. At worst, system latency can cause simulator sickness and make the devices unpleasant to use. In these devices there are multiple different components that together construct the final system latency of the device. These components are tracking delay, application delay, image generation delay and display delay. The image generation delay and display delay can also be described as see-through or visual latency. It is an latency that is only present in mixed and augmented reality devices due to their ability of mixing virtual and real environments. See-through latency is the delay between the time of the device observing the real world environment around the user and based on this, generating imagery on the device display based on the user input.

At the time, there isn't any standard way of measuring see-through latency in mixed reality headsets. In this thesis a measurement method based on hardware instrumentation is presented. Two different approaches to the see-through latency measurement are described and analyzed. The first approach is based on modifying color sensor signal from which the final see-through is derived from. The signal is low-pass filtered to remove the backlight of the tested device's display. This done to decrease the amount of false positives of detected see-through events from the color sensor data. In the second approach, the color sensor signal itself isn't modified, but the detected see-through events are gathered in a different way. The color sensor data is converted into brightness values, and a high envelope of the signal is derived. The signal is then divided into frames and from these frames the local maxima can be gathered. If the brightness of this data point is over a certain threshold, it will be classified as a blinking event and it will be used in calculating the final latency.

The measurement methods described gave quite good results. Because the latency is highly dependant on the tested device, the measurement might give better results for some devices and worse results for others. The first measurement method couldn't always remove the backlight of the tested device as a see-through detection, which resulted in a large amount of false positives. This resulted in a lower final latency value than it should be. With the second measurement method these false positives could be eliminated and the measured latency values were more accurate due to the more universal method of deriving the see-through events.

Keywords: Extended Reality, Mixed Reality, Head-Mounted Displays, Latency

TIIVISTELMÄ

Inka Juntunen: Visuaalinen latenssi sekoitetun todellisuuden laitteistoissa
Kandidaatintyö
Tampereen yliopisto
Sähkötekniikan tutkinto-ohjelma
Joulukuu 2022

Laajennettu todellisuus ja erityisesti sekoitettu todellisuus ovat kasvussa olevia teknologian aloja, joilla on useita hyödyllisiä sovelluskohteita. Niitä ovat esimerkiksi viihteen sovellukset, kuten pelaaminen, koulutus ja opetus, sekä lääketieteen sovellukset. Koska nämä sovellukset vaativat laitteistolta tarkkuutta, laitteen käyttöön liittyviä parametreja halutaan saada mitattua mahdollisimman tarkasti.

Sekoitetun todellisuuden laitteistoissa esiintyvällä latenssilla on suuri vaikutus laitteen käytettävyyteen. Pahimmassa tapauksessa systeemin latenssi voi aiheuttaa simulaattorisairautta ja tehdä laitteistosta epämiellyttävän käyttää. Näissä laitteistoissa systeemin latenssi koostuu muutamasta eri tekijästä: seurantaviiveestä, sovellusohjelman viiveestä, kuvantamisen viiveestä ja generoidun sisällön esittämisen viiveestä. Sovellusohjelman, kuvantamisen ja generoidun sisällön esittämisen viivettä voidaan kuvata myös termeillä läpinäköviive tai visuaalinen latenssi. Se on latenssimuoto, jota on vain sekoitetun ja ehostetun todellisuuden laitteistoissa, koska näillä teknologioilla on mahdollista yhdistää oikeita ja virtuaalisia ympäristöjä. Visuaalinen latenssi kuvaa aikaa joka laitteistolla menee, jotta se saa reaalimaailman ja mahdollisen ehostuksen esitettyä laitteiston näytöllä ja käyttäjä pystyy näkemään muodostetun syötteen.

Tällä hetkellä ei ole standardoitua tapaa mitata visuaalista latenssia. Tässä kandidaatintyössä esitetään mittausperiaate, joka perustuu ulkoisen laitteiston käyttämiseen latenssimittauksen suorittamisessa. Kahta hieman toisistaan eroavaa mittaustapaa tarkastellaan ja niistä saatuja tuloksia analysoidaan. Ensimmäisessä mittaustavassa ulkoisen laitteiston värisensorista saatavaa dataa muokataan alipäästösuodattamalla muodostettua kirkkaussignaalia. Tämä tehdään, jotta testilaitteiston näytön taustavalosta johtuvat väärät positiiviset havainnot saadaan poistettua lopullisista luokitelluista tapahtumista. Toisessa mittaustavassa latenssi määritetään raakadatan avulla. Sensoridatan perusteella lasketaan havaittujen tapahtumien kirkkaudet ja määritetään signaalin verhokäyrä. Tämän jälkeen signaali jaetaan ruutuihin, ja näiden ruutujen sisällä olevat maksimiarvot määritetään. Jos nämä arvot eroavat tarpeeksi ennalta määritetystä alarajasta, merkataan arvot havaituiksi tapahtumiksi ja niiden avulla lasketaan laitteiston lopullinen visuaalinen latenssi.

Molemmista mittaustavoista saatiin käyttökelpoisia tuloksia. Koska mitattu latenssi on hyvin riippuvainen testatusta laitteistosta, mittaus saattaa antaa joillekin laitteistoille parempia tuloksia kuin toisille. Ensimmäisessä mittaustavassa ongelmaksi muodostui testattavan laitteen näytön taustavalon tehokas poistaminen. Siihen ei aina pystytty, joka johti suhteellisen suureen määrään vääriä positiivisia havaintoja. Tapahtumia siis merkattiin havaituiksi, vaikka niitä ei ollut. Tämän vuoksi lopullinen latenssiarvo saatettiin mitata matalampana kuin mitä se todellisuudessa on. Toisena esitetyllä mittaustavalla saatiin poistettua nämä aiemmin esiintyneet väärät positiiviset havainnot yleiskäyttöisemmän tapahtumien luokittelutavan vuoksi. Tämän vuoksi toisena esitetyllä mittaustavalla saatiin myös todenmukaisempia ja tarkempia visuaalisen latenssin mittausarvoja.

Avainsanat: Laajennettu todellisuus, Sekoitettu todellisuus, Kypäränäytöt, Latenssi

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

AR	Augmented Reality
DOF	Degrees of Freedom
DUT	Device Under Testing
FOV	Field of View
HMD	Head-Mounted Display
Latency	Time delay between cause and effect in a physical system
MR	Mixed Reality
OST	Optical See Through
VR	Virtual Reality
VST	Video See Through
XR	Extended Reality

1. INTRODUCTION

Extended reality (XR) is an emerging technology with a wide range of many beneficial applications, which include gaming, education and training and medical applications. Extended reality is an umbrella term used to describe all different technologies related to each other. [1] These technologies are virtual reality (VR), augmented reality (AR) and mixed reality (MR). The three technologies are very similar to, but there are enough differences to differentiate them into their own categories. In this thesis the focus is on mixed reality, which can shortly be described as a mixture of virtual and augmented reality. It is an technology where the user can be fully immersed in a virtually created world while at the same time being capable of interacting with the real environment around them. Extended reality technology devices are usually implemented as head-mounted displays (HMDs), which are placed on the user's head when in use. [2]

System latency is a crucial thing that is highly influential to the user experience of mixed reality systems. If there is too much latency in the device, it impacts user performance and makes the device uncomfortable to use and can cause simulator sickness, which is an unwanted side effect. [3] This is the reason why manufacturers want a way to measure system latency as precisely and efficiently as possible. One cause of system latency in mixed reality devices is see-through latency, which describes the time delay of image generation from detected imagery from the real environment to the recreation of it to the HMD display. At the time there is no standard way of measuring see-through latency. [4] In this thesis, two different measurement methods created at OptoFidelity are presented and compared to each other and a few alternative measurement methods presented in chapter 2.3.

The purpose of this thesis is to describe the implementation of two different see-through latency measurement methods and analyze their results. The accuracies of the presented see-through latency measurement methods are also compared to each other to see the possible differences in the methods and therefore see which one is more suitable for future use.

The remainder of this thesis is structured as follows: Chapter 2 presents the required background information on mixed reality technology, the cause and effects of system latency and a few alternative methods of measuring visual latency. Chapter 3 contains the

see-through latency measurement principle created at OptoFidelity, including the implementation of the two different measurement methods, analyzing their results and comparing them to each other and other alternative measurement methods. Finally, in Chapter 4 the thesis's conclusions are presented.

2. BACKGROUND

2.1 Mixed Reality

Extended reality is an umbrella term that can be used to describe virtual reality, augmented reality and mixed reality technologies. They are all immersive technologies that can be applied from gaming to medical applications and training. [1] Virtual reality (VR) refers to the technology where realistic virtual environments are created, and the user can interact with this virtual reality in real-time. The user can not access the real world environment around them while using VR technology. Augmented reality (AR) allows the user to see the real world with augmentation or overlaid virtual elements. For a device to be considered an AR device, it needs to fulfill the following criteria according to Azuma [5]: the system has to combine real and virtual objects in a real environment, it has to run interactively, and register or align real and virtual objects with each other [6]. The technology most present in this thesis, mixed reality (MR), can be described as a mixture between VR and AR. MR is the term used to describe the relationship between reality, virtual reality, augmented reality and augmented virtuality. The relations of these stages can be described with the virtual continuum presented below in figure 2.1. [2]

Figure 2.1. *The virtual continuum [7], adapted from [2]*

Due to the fact that with mixed reality, the user can be fully immersed and yet at the same time interact in social situations and with real-world objects makes this technology interesting for a wide range of applications [1]. These include military applications, entertainment and infotainment, technical support, industrial applications [1], education and training [8], and medical use [9]. In education and training, MR can be used, for instance, visualizing digital 3D models and their exploded diagrams to improve the student's comprehension of the designs and understanding of how the design object functions [8]. In the field of industrial applications MR can be applied, for example, to architecture and construction. MR can be very useful for visualizing designs, on-site construction planning and as-built verification, where the achieved built quality is compared to the one aimed for [10]. In the medical field MR can be used in medical education, training, rehabilitation, surgical planning and as a guide in complex medical procedures. Out of these medical applications, education and rehabilitation are the most appealing options to invest in. These areas don't require direct involvement with patients, so there will be no controversial ethical questions to take into consideration. [9] The range of possible use cases for mixed reality head-mounted displays is quite wide, and the technology is promising. Despite of this, at the time, there aren't many options for off-the-shelf practical and available MR systems created for anyone other than the research community [1].

2.2 Mixed and Augmented Reality Display Technologies

To understand the see-through latency measurement better, it is important to know about the different available display technologies for extended reality (XR) head-mounted displays (HMDs). The available technologies can be roughly divided into two categories, video see-through (VST) and optical see-through (OST) HMDs [11].

With video see-through technology (VST), the user can't see the real world around them when the see-through mode is off, and the HMD isn't reflecting the user's surroundings on it's display. This means the user is in a closed environment when using the device. In VST HMDs, the user's surroundings are captured using cameras, and then the viewed environment is generated with software to the display of the device. [11] An example of a VST HMD device structure is shown below in figure 2.2.

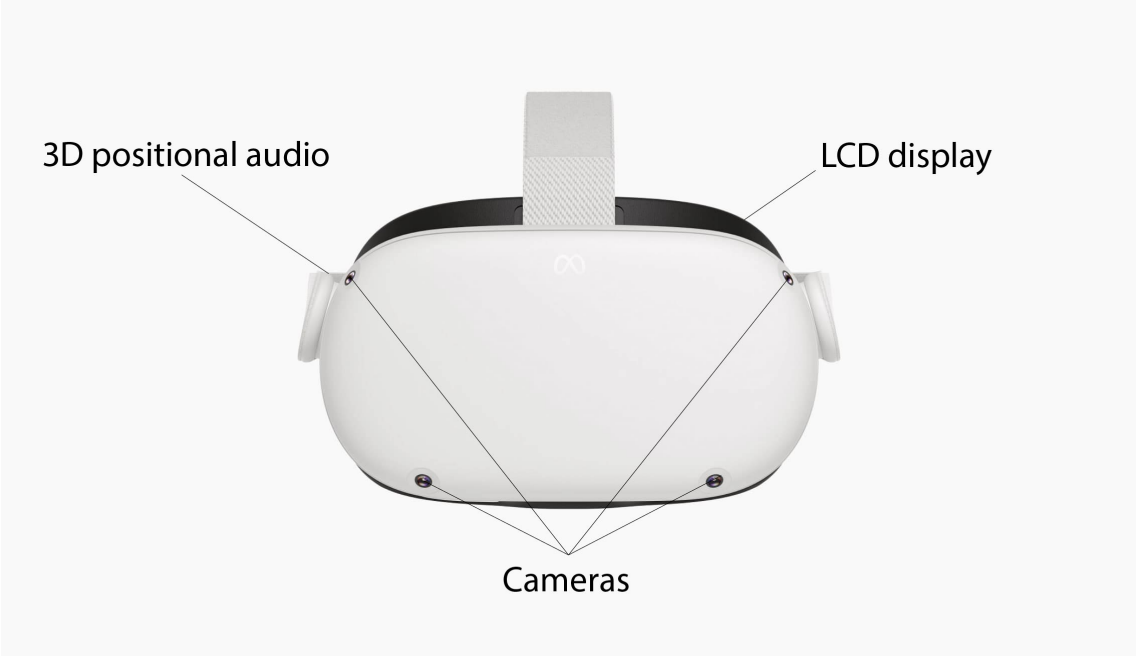

Figure 2.2. *Meta Quest 2 Video See-through Head-Mounted Display [12]*

When the surrounding environment is entirely computer generated, it causes some problems which limit the possibilities of using this technology. These issues are related to focusing and the limited depth of field. These problems occur because the human eye viewing the display can change its focusing distance through accommodation, unlike the cameras in the device. Accommodation affects the depth of field, so when the cameras don't act similarly to the human eye. This results in an unrealistic user experience and warps the depth perception of the user. [13] Especially in medical applications, false depth perception is a crucial problem. When this happens, the user is unable to reach or manipulate real world or virtual objects coherently, which could in the worst case result in endangering patient care. [14].

In optical see-through (OST) HMDs, the augmentation is done by reflecting computer generated graphics to the semi-transparent displays placed in front of the user's eyes. Because of these semi-transparent displays, the user can see the environment around them as is, unlike in VST HMDs. [15] Because of this, OST technology can't be used in VR applications, like the VST technology presented above [16]. For OST, there are a few different options for the display technology. Some general methods used for augmentation are half-mirrors, birdbath combiners, free-form prisms, waveguides, retinal scanning, liquid crystal display (LCD) layers, microlens arrays, pinlight displays and 3D holographic displays [15].

In the first OST HMD device referred to as "Sword of Damocles" half-silvered mirrors were used to reflect imagery into the users view. These semi-transparent mirrors were then placed in from of the user's eyes and the image was shown on a small cathode-ray

tube. Most OST technologies following this first system are still based on this primitive concept. A small improvement compared to the half-mirror mirror design is made with birdbath optics, where a curved lens is combined with the half-mirror, therefore achieving a larger field of view (FOV). It's not a perfect solution because with this approach, the issue of light efficiency is still present. This happens because the light travels through the half mirrors multiple times. The added curved lens makes the design also larger and heavier than with half-mirror technology. This leads to an uncomfortable device to wear and use. In newer prototypes, these half-mirrors are replaced with other optical elements which are lighter and have better light efficiency. One option is to use optical prisms or free-form optics which have similar display results as half-silvered mirrors, but with better light efficiency. The problem with optical prisms is that they need be to quite thick, which makes the HMD heavy, and, again unpleasant to use. [15]

A better option compared to the optic solutions presented in the previous chapter are waveguides, which can be used, for instance in retinal projector displays or retinal scanning and microlens array technologies. [15] Waveguide technology is based on emitting a computer-generated image and outside light in the waveguide and using its internal reflection to guide the emitted light toward the user's eyes. Waveguides are not only thinner and lighter than optical prisms. They also increase the FOV, have high light efficiency and have a simple fabrication method which leads to a lower manufacturing cost compared to other available options. The problem with only using waveguides for the augmentation is that it leads to ghost images. These ghost images can be eliminated by adding a micro-mirror or microlens array to the waveguide. [17]

Another improvement in waveguide technology is retinal scanning. In that, thin light beams are emitted from a semiconductor laser and transmitted through optical fibers to a scanning micro-electro-mechanical-system, or MEMS mirror. Then this mirror scans the thin laser beam in two directions of parallel light. This light then hits the vertical surface of the holographic film on the inner side of the waveguide and diffracts. This diffracted light is then guided through the waveguide. Another holographic film is attached to the outer side of the waveguide, which diffracts the light to a harmonic lens. Finally all the diffracted scanning beams are focused, and an image is created for the human eye through a harmonic lense. [18] An example of a OST HMD using waveguide technology and its hardware specification is presented below in figure 2.3 [19].

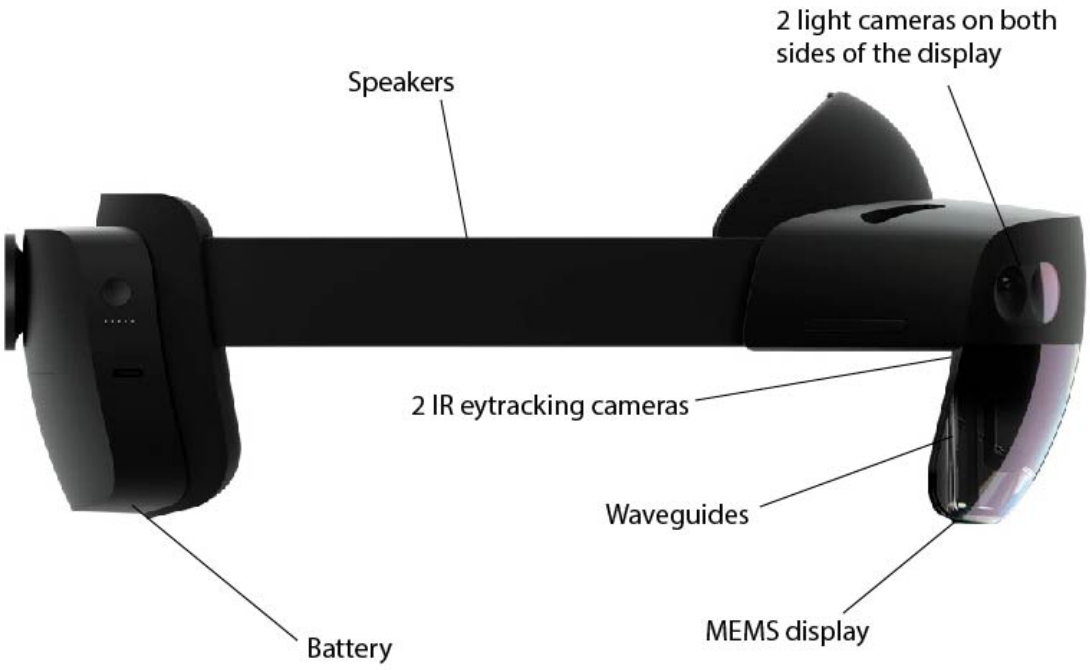

Figure 2.3. *Microsoft HoloLens 2 Optical See-through Head-Mounted Display [20]*

Another possible approach is to use multiple stacked liquid crystal displays (LCDs) for image formation. For instance, in a pinlight display, the light source is modified by placing an array of point light sources behind an LCD which creates a pinhole projection that is always in focus. In addition to these technologies, a 3D holographic display can be used for image formation without LCDs. [15]

The extended reality industry and researchers have shown more interest and made investments in the OST HMD technology in recent years. This is because with OST technology the HMD can be made to resemble regular glasses and by that showing the environment at its highest fidelity, unlike with the VST technology. Both of these technologies have their advantages. With VST technology, the creation of the virtual world is easier because the camera feed can be visually merged with the virtual environment in the camera-image space. On the other hand, while using the device, the user is disengaged from reality and can't interact with other users in the same space in the same way as with OST technology. The challenge with OST technology is creating convincing augmented or mixed reality experiences. [15] The issue that needs improving with these both display technologies is the depth ordering and perception problems. As mentioned earlier when discussing the VST HMDs, human eyesight perceives depth with depth cues, and if the depth perception is distorted, interaction is affected and the immersion is lost. [14] Based on these things, OST technology seems like a more promising technology for future HMD

devices, although there still is many things to be tackled before these devices can be commercialized and introduced to the masses. The devices need to be miniaturized, accomodation and latency issues need to be fixed, FOV needs to be increased, and the colour representation and resolution issues need to be addressed. [15]

2.3 Latency in Head-Mounted Displays

Latency in see-trough head-mounted displays can be described as the time from the user input, which could be head motion or imagery from the HMD cameras, to the corresponding feedback on the updated HMD display. If there is apparent latency, it can be perceived by the user, causing simulator sickness. Simulator sickness can distract the user and be the cause of poor performance at the task at hand, and reduce user acceptance of the device. [3] In AR and MR HMDs latency can result in virtual and real objects moving relative to one another, which is called slipping. [4]

System latency is usually the combination of tracking delay, application delay, image generation delay and display delay [15]. The tracking delay can also be described as motion-to-photon latency [21], and the combination of application delay, image generation delay and display delay as photon-to-photon or see-through latency [4]. The former is related to the head movement of the user. It describes the latency of the device's tracking capability of the user's head movements. [21] The latter describes the time that the device takes until the real-world imagery reaches the user's eyes [4]. In this thesis focus is on the see-through latency.

In video see-through AR and MR systems the see-through latency is the sum of multiple components in the system. It contains exposure time of the cameras, image transfer, tracking, visualization, and finally, displaying the image. Exposure time, transferring the image to the device memory, and the time the final image remains on the display can be obtained from the hardware. Other steps are computer generated, and their latencies can vary based on the methods used. [22] The human body also has an intrinsic latency in its sensory system, which depends on the input modality. Visual stimuli are perceived at the neuronal level after roughly 200 ms from the stimulus. Auditory signals can be detected within 100 ms of the stimulus. In addition to these latencies, additional delay needs to be added for the motor response. Based on this, can be assumed that after perceiving the stimuli, the human response time to the used stimuli would be between 300 and 400 ms. If response time is over this level, it can be expected to be the result of other system delays that are not related to human processing the incoming signals. Gruen et al. have introduced a latency measurement method based on these human response times. The method is inspected in more detail in the last chapter. For VR and the Oculus systems, the suggested maximum system latency is less than 20 ms. In AR and MR, and especially in optical see-through, the system latencies need to be less than 1 ms to maintain the

registration between virtual and real objects. For instance, in touch pointing tasks, have been discovered that the users can identify 1 ms latency and 10 ms starts to affect their performance. [4]

There is no standard way of latency measurement of an AR or MR system at the moment. [22] The most common approach is hardware instrumentation based, where an external reference is used to monitor physical movement or change in the reference. One way of doing the measurement is by taking a video of the magnetic tracker on a pendulum together with a screen and at the same time, showing a timestamp of the latest tracking information. From reconstructing the pendulum motion and finding the lowest point of the pendulum from the video, the latency can be approximated. One effective measurement method is to use two photodiodes to measure the pendulum's lowest points, and then determining the time difference with an oscilloscope. One photodiode tracks the low point of the real pendulum, and the second one tracks the rendered version of the tracked motion. [4] In addition to the previous approaches, Sielhorst et al. have introduced a latency measurement based on camera feedback. The latency estimation is done by encoding the time in the image, and after camera feedback the time is decoded. This type of measurement had adequate accuracy and the advantage of not needing any additional hardware for the measurements, but the measurement would have to be built in to the device itself. This means that the manufacturers would have to implement the latency measurement system in addition to the actual functionality of the the device. [22]

A differing approach was introduced by Gruen et al., where the latency measurement was not based on hardware instrumentation but on human cognitive performance. In this method, the test subjects were wearing a VST HMD device and had them press a sizeable physical button in front of them when they viewed a rendered circle on a computer screen. The system latency was then derived from the time difference between rendering the circle and the test subject triggering the button signal. This measurement is based on the assumption that human performance remains constant, and any added latency equals to the system latency. [4] This cognitive latency approach could be only used in VST HMDs as is [4], but the camera feedback method could be used in both VST and OST HMDs [22].

2.4 OptoFidelity Buddy

OptoFidelity's Buddy is a AR, VR and MR HMD testing device that can be used for calibration and measuring system performance. Buddy has an integrated vision module and either 3 or 6 degrees of freedom (DOF). The DOF selection based on the device under testing (DUT). Buddy can be used for measuring the HMDs motion-to-photon latency, content stability and pose drifting. The system can measure both present and future display systems, for instance OLED and projector based, holographic and light field systems.

[23] The main elements of the system are presented below in figure 2.4.

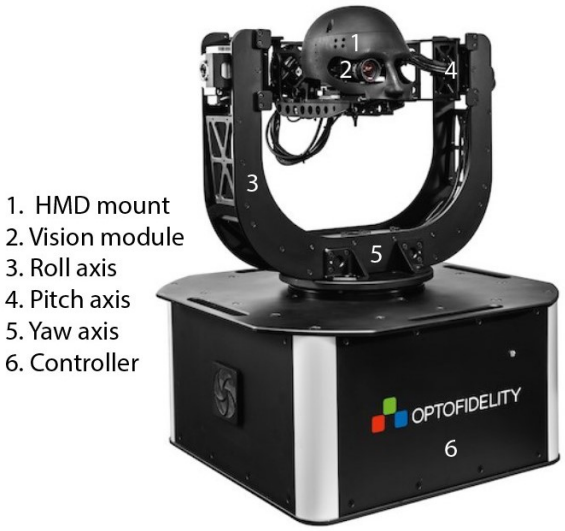

Figure 2.4. *OptoFidelity Buddy [24]*

Buddy's measurement system is based on non-intrusive measurements, where changes in the robots virtual or augmented pose is compared to the changes in the real environment pose. [24]

3. SEE-THROUGH LATENCY MEASUREMENTS

3.1 Measurement principle

The see-through latency measurements are implemented at OptoFidelity premises, and some information is confidential. Everything that can be disclosed about the measurements will be presented.

The see-through latency measurement setup consists of a see-through target that is attached in front of the tested HMD, a device under testing (DUT), Buddy, and a PC for the latency calculations. Buddy has two cameras and a color sensor in its vision module. Either one or both cameras can be used to do the measurement. [25] The basic principle of the latency measurement is shown in figure 3.1 below.

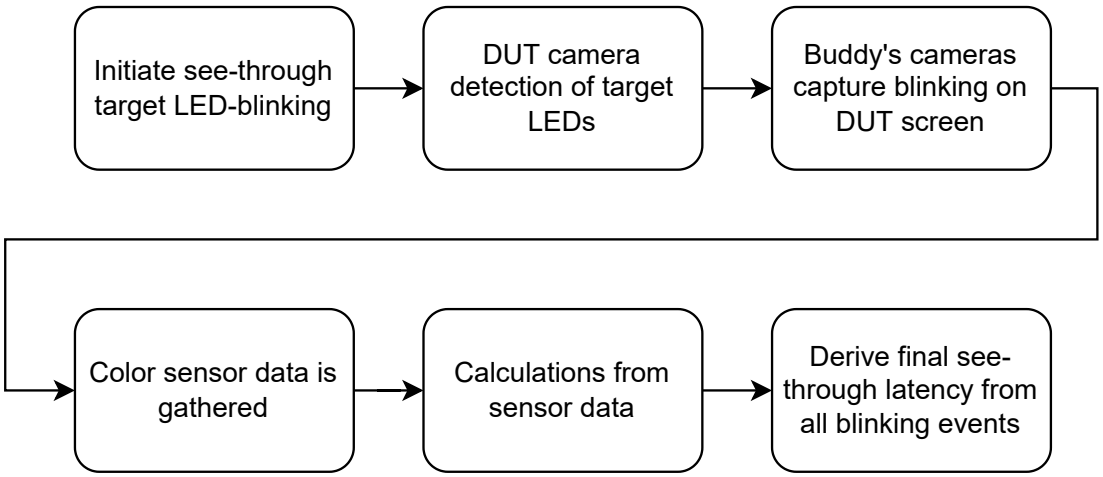

Figure 3.1. *See-through latency measurement principle*

The first step of the measurement is initiating the blinking sequence of the see-through target, as can be seen from figure 3.1. The see-through target consists of LEDs with an RGB color range and are blinked at the desired interval I. The DUTs used in all the measurements presented are VST type, so the device will detect the blinking of the see-through target LEDs with its cameras. When the DUT cameras detect the blinking, it recreates the same image to its display as seen from the cameras. This image formation is the process that creates the see-through latency that is measured. After the LED blinking can be seen on the DUT display, Buddy's cameras or camera detect the blinking

and color sensor data is received in XYZ form. The data is then transferred to the PC for computation. Then the brightness of the DUT display can be calculated from the color sensor data. [25]

The calculation method of the brightness varies depending on the used implementation. The blinking events of the DUT display can be derived based on the calculated brightness values. If the brightness is over a certain threshold, it will be considered a blinking event, and its timestamp T_n is saved. The see-through latency for each blinking event can then be calculated as the remainder of the used interval

$$L_n = T_n \mod I \tag{3.1}$$

where L_n is the see-through latency of a specific event, T_n is the timestamp of the detected blinking, and I is the interval used for blinking the target LEDs. With this equation, the largest possible latency is the blinking interval, and the final see-through latency of the tested HMD is the average value of all event latencies. [25] This average latency can be calculated with

$$L_{ave} = \frac{\sum_{n=1}^{n} L_n}{n} \tag{3.2}$$

where L_{ave} is the average of all blinking event latencies, L_n is the latency of a specific blinking event and n is the total number of detected blinking events. In addition to the average latency, the standard deviation of the latencies will be calculated to see how much the latency values deviate from each other. The standard deviation can be calculated with the following equation

$$\sigma_L = \frac{\sum_{n=1}^{n} (L_n - L_{ave})^2}{n} \tag{3.3}$$

where σ_L is the standard deviation of the latencies, L_{ave} is the average of all blinking event latencies, L_n is the latency of a specific blinking event, and n is the total number of detected blinking events. In reality, the calculation of average latency and standard deviation of detected blinking events is implemented in the programming language Python by using NumPy library's numpy.mean [26] and numpy.std [27] functions, which are based on the presented equations 3.2 and 3.3.

To eliminate Buddy's system latency from the final see-through latency, the measurement is first done without the DUT. This latency corresponds to the system latency and is subtracted from the see-through latency results.

3.2 Deriving see-through latency from processed sensor data

In the first achieved measurement method, the see-through latency was derived from processed color sensor XYZ-data. The brightness value of the DUT display is obtained by averaging the signal value over its XYZ-axes. Then the signal is low-pass filtered to re-

move the effect of the DUT backlight, which might otherwise appear as an blinking event. This would cause false detections in the final calculations. If the calculated brightness values are over the average brightness value by a predetermined threshold, the event will be marked as a positive blinking event, i. e. marked as the DUT capturing the see-through target LED blinking. Positive blinking events are marked as 'w' events and negative ones as 'k' events. The final see-through latency is calculated from the timestamps marked as a 'w' event. [25]

The see-through latency measurements were done with two different DUTs, first with the approach of using processed color sensor data. DUT_1 is an Oculus Quest 2, and DUT_2 is a Pico Neo 3. Figures 3.2 and 3.3 present the color sensor data from Buddy's vision module cameras CAM_1 and CAM_2. In both figures, the detected blinking event is represented on the x-axis and the latency of the event is represented in the y-axis in milliseconds. The measurement period is 30 s. [25]

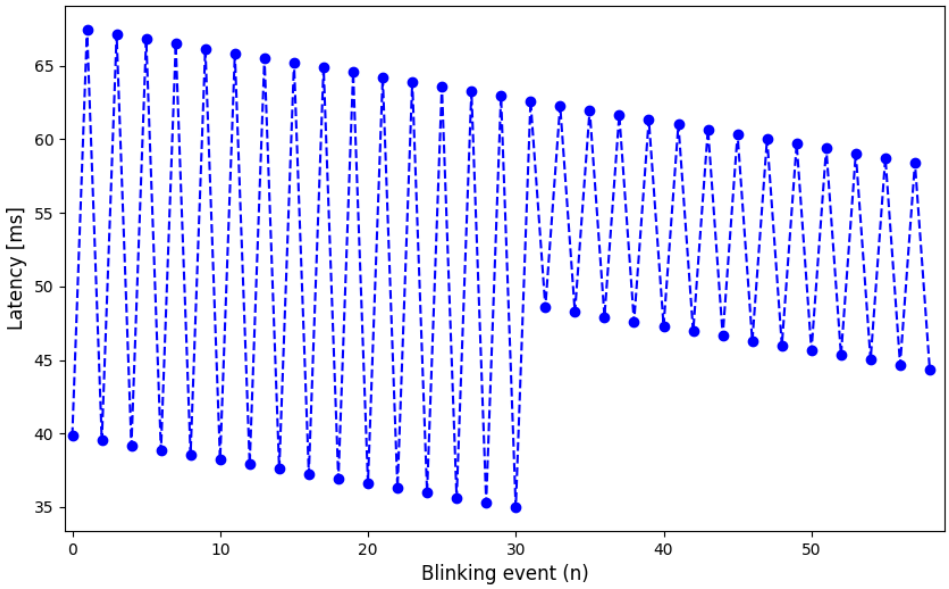

Figure 3.2. *See-through latency of DUT_1 using CAM_1, adapted from [25]*

The see-through latency 'w' events of CAM_1 can be seen in figure 3.2 x-axis. The amount of blinking events depends on the chosen blinking interval I. As can be seen from figure 3.2, the latency value fluctuates between the minimum latency of 34.998 ms and the maximum latency of 67.444 ms. The measurement sequence has 59 detected events in total, and from 3.2 and 3.3 the average latency and standard deviation for DUT_1 using CAM1 can be calculated. Now $L_{ave_{1,1}} = 52.112$ ms and $\sigma_{L_{1,1}} = 11.327$ ms. [25] The see-through measurements are also done for DUT_1 with camera CAM_2 and its measurement results are presented below in figure 3.3.

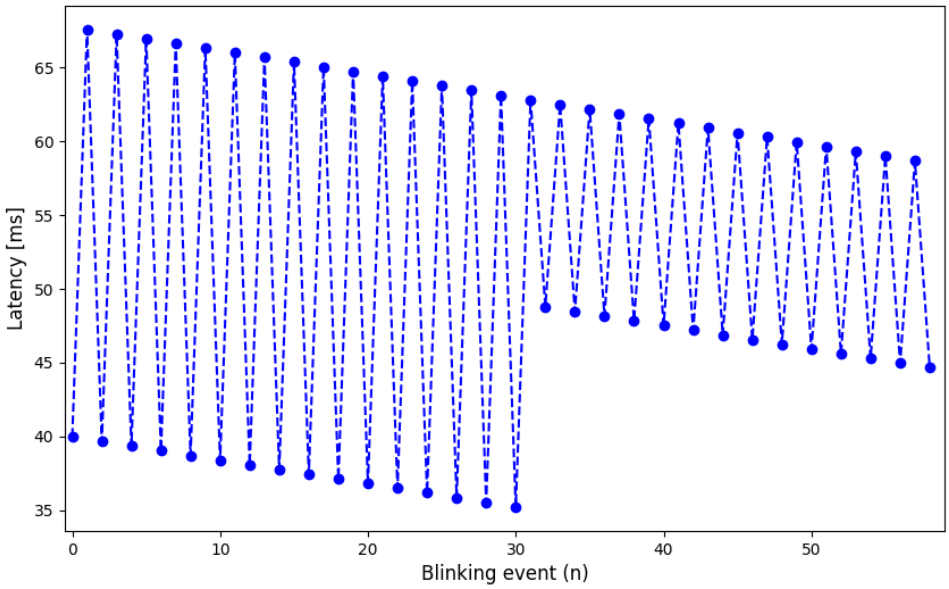

Figure 3.3. *See-through latency of DUT$_1$ using CAM$_2$, adapted from [25]*

As can be seen from figure 3.3, the measurement results are very similar to those presented in figure 3.2. With camera CAM$_2$ the measured minimum latency is 35.321 ms, and the maximum latency is 67.582 ms. The measurement sequence has 59 detected events in total, and the average latency and standard deviation can be calculated with equations 3.2 and 3.3. The average latency and standard deviation with CAM$_2$ is $L_{ave} = 52.321$ ms and $\sigma_{L_{1,2}} = 11.330$ ms, so there is only a $L_{diff_1} = |L_{ave_{1,2}} - L_{ave_{1,1}}| = 0.209$ ms difference with this measurement sequence when measuring average latency with CAM$_1$ and CAM$_2$. [25]

See-through latency measurements were also done with DUT$_2$. Measurement results for DUT$_2$ latency using CAM$_1$ are presented below in figure 3.4. The measurement period is again 30 s, and the blinking event latencies in milliseconds are presented as a function of blinking events.

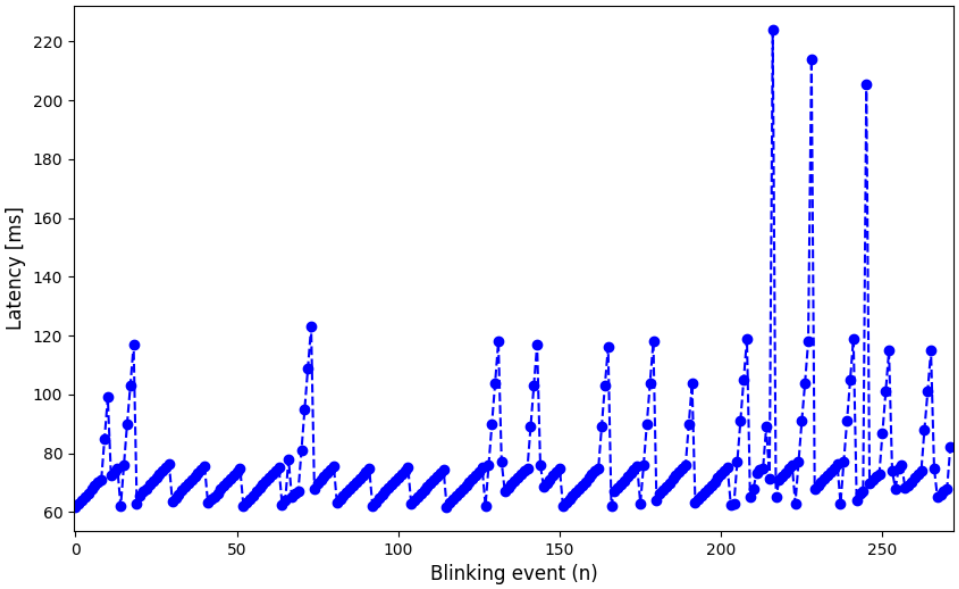

Figure 3.4. *See-through latency of DUT$_2$ using CAM$_1$, adapted from [25]*

As can be seen from figure 3.4, now with DUT$_2$ and CAM$_1$ there are 271 detected blinking events, and the see-through latency is 224.000 ms at maximum and 61.701 ms at minimum. The average latency and standard deviation can be calculated with equations 3.2 and 3.3. Now $L_{ave} = 74.430$ ms and $\sigma_{L_{2,1}} = 12.826$. [25] The measurements were done simultaneously for CAM$_2$, similar to the previous measurement for DUT$_1$. The measurement results for DUT$_2$ latency using CAM$_2$ are presented below in figure 3.5.

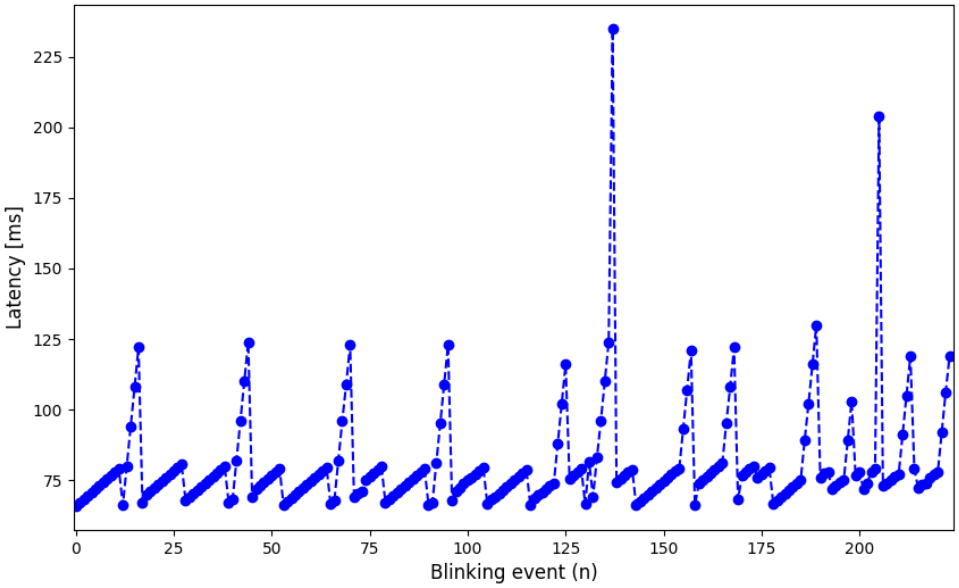

Figure 3.5. *See-through latency of DUT$_2$ using CAM$_2$, adapted from [25]*

With CAM_2 there are 223 detected blinking events in total. As can be seen from figure 3.5, the minimum latency value for CAM_2 is 61.701 ms, and the maximum value is 235.000 ms. The average latency and standard deviation can be calculated in the same way as previously, with equations 3.2 and 3.3 we get $L_{ave_{2,2}} = 79.302$ ms and $\sigma_{L_{2,2}} = 13.695$ ms. The difference between the average latencies with the two cameras for DUT_2 is $L_{diff_2} = |L_{ave_{2,2}} - L_{ave_{2,1}}| = 4.872$ ms. [25]

3.3 Deriving see-through latency from raw sensor data

Another approach for the measurements is to use raw sensor data for the calculations. This means that there is no post-processing involved before the calculations, like in the method described in chapter 3.2 where low-pass filtering was used to eliminate the DUT's display backlight. With this second method, the brightness values of the blinking events are calculated from the color sensor XYZ-data by taking the Euclidean norm of the XYZ-vector with the following equation

$$Brightness_{XYZ} = \sqrt{X^2 + Y^2 + Z^2} \tag{3.4}$$

where $Brightness_{XYZ}$ is the brightness value and X, Y and Z are the obtained axis values. The signal is divided into frames to eliminate other color sensor activity than the detected blinking in the DUT display. First the upper envelope, or an outline of signal extremes, is derived from all the local maximum brightness levels of the signal. Then the chosen frame is moved forward by a predetermined hop size. Then from these individual frames, the maximum brightness inside each frame can be determined. If the difference in the brightness levels inside consecutive frames is over a predetermined threshold times the global maxima of the signal found from the upper envelope, a blinking event is detected. This way, other DUT display activity can be eliminated, and the latency measurement should become more accurate. [25] This measurement method is still under development at the time of writing this thesis, and at hand is only measurement data for DUT_2 with CAM_1. Below in figure 3.6 are represented the latency results as a function of event timestamps for DUT_2 using CAM_1.

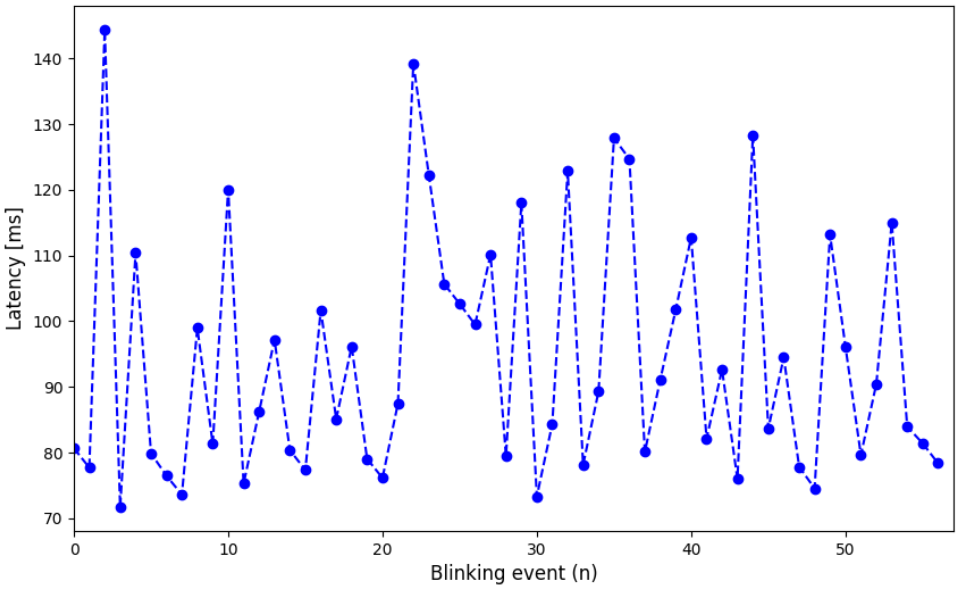

Figure 3.6. *See-through latency of DUT$_2$ using CAM$_1$, adapted from [28]*

With the raw data method, 56 blinking events are detected, with maximum see-through latency of 144.374 ms and minimum latency of 71.712 ms. Average see-through latency value and standard deviation for DUT2 using CAM1 can be calculated with equations 3.2 and 3.3. Now average latency is $L_{ave} = 94.429$ ms with a standard deviation of $\sigma_L = 18.529$ ms. [28]

3.4 Results and analysis

The average see-through latencies derived with the processed sensor data for Oculus Quest 2 with camera CAM$_1$ is $L_{ave_{1,1}} = 52.112$ ms and with camera CAM$_2$ $L_{ave_{1,2}} = 52.321$ ms. For Pico Neo 3 the measured latency with camera CAM$_1$ is $L_{ave_{2,1}} = 74.430$ ms and with camera CAM$_2$ $L_{ave_{2,2}} = 79.302$ ms. Figures 3.2 and 3.3 show that the individual latencies remain stable in both measurements with CAM$_1$ and CAM$_2$, and there are no outliers in the test sequences. The latency results are also very close to each other with only a 0.209 ms difference. Based on this, the test results for the Oculus headset are reliable and resemble the true see-through latency of the system.

For the Pico headset, the measurement isn't as accurate as for the Oculus Quest 2. From 3.4 and 3.5 can be seen that there are many false positives in the detected blinking events. With both DUTs the measurement period was 30 seconds, and the LED blinking interval I was the same. With the Oculus Quest 2, there were only 59 detected events with both cameras, while with Pico Neo 3 there were 269 and 222 detected events with the same measurement parameters. If the see-through target blinking is the same in

both measurements, the number of events detected should remain roughly the same too. The false positives in figures 3.4 and 3.5 can be seen as the steadily raising edges at low latencies, which form a saw blade-like pattern in the figure. That is the result of the DUT backlight being detected as a blinking event, which the low pass filter cannot eliminate in this case. When these events are then used for the latency calculation, the final see-through latency is lower than it should be due to a large amount of low-latency components in the computation. The final latencies for the Pico Neo 3 with CAM_1 was $L_{ave_{2,1}} = 74.430$ ms and with CAM_2 $L_{ave_{2,2}} = 79.302$ ms. The difference between these two latencies is 4.872 ms, which is significantly larger than the difference between the measured see-through latencies with Oculus Quest 2.

The see-through measurement is highly dependable on the used DUT. This is because many variables that affect the measurement are unpublished by the manufacturer. The things that impact the measurement the most are the camera capture frequency, camera capture time and the backlight brightness. The camera parameters need to be considered when making the see-through target and when blinking the LEDs. If the blinking interval doesn't match the capture frequency, the blinking could be missed entirely. If the LEDs in the target are blinked in an undesirable way for the DUT, again, the blinking could be missed. The DUT backlight affects the detection of the see-through target blinking, and may cause false positives for some devices. Because of this, the measurement method needs to be as universal as possible for it to be used on a wide variety of MR HMD devices. When using a low pass filter, the DUT backlight could not always be removed from the detected events. This is the reason another method was created for the see-through latency measurement, which should eliminate the false positives from the detected events, which in this case, is the DUT backlight.

At the time of writing this thesis, there is only test data available for the raw color sensor data approach. The measurement is done for Pico Neo 3 with only camera CAM_1. From the figure 3.6 can be seen that the false positives are now eliminated, and there are now only 56 detected blinking events, and the average latency is $L_{ave} = 94.429$ ms. These results would indicate that the raw data method is more capable of identifying false positives from the sensor data than the first approach. This improved performance is based on the fact that the see-through target LED blinking interval I is known. From this, a suitable frame size can be chosen, meaning that the frame is sized in a way that there is only one blinking event in it. When the maximum value inside this frame is derived and then compared to a threshold, it will eliminate blinking detections that are not bright enough to be a see-through event.

The measurement results for all test cases are summarized in the violin plot 3.7 below. The middle line of the figure objects show the mean value of the measurement, and the object curves display the approximate frequency for specific latency values in the measurement. Above each measurement its mean value is displayed.

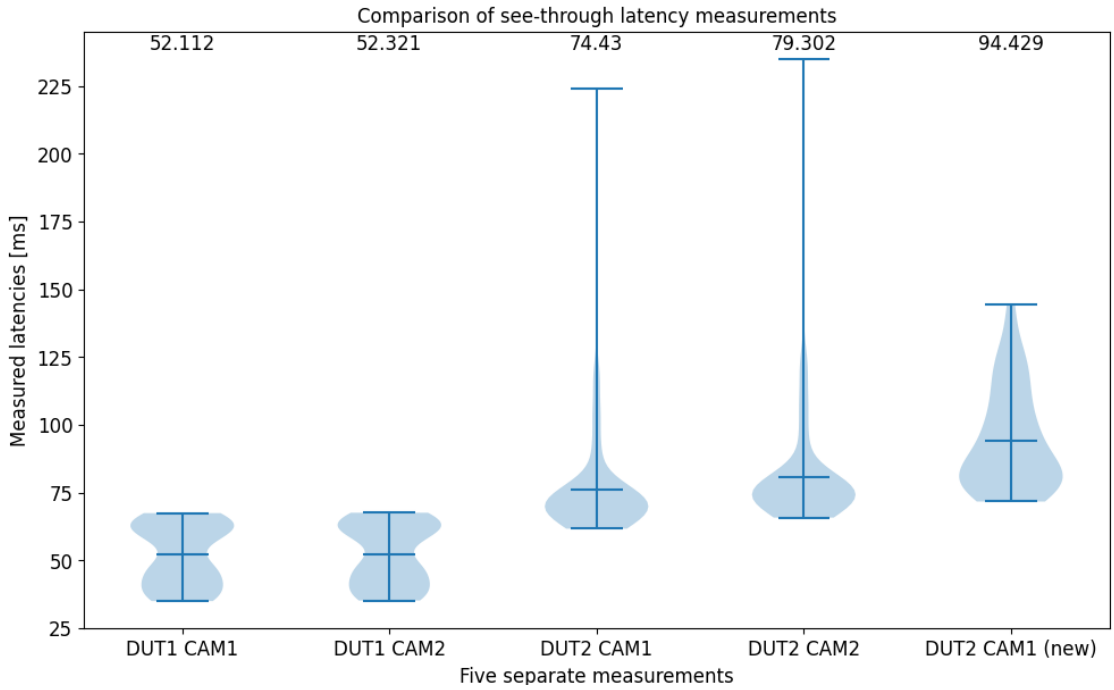

Figure 3.7. *Summary of all test results [25]*

As can be seen from figure 3.7, the test results for DUT_2 differ from each other between the two test methods. With the old measurement, most of the density in the plot is located at the 75 ms area. Also, the difference between the largest and smallest measured latency value is bigger than in the other measurements. With the new measurement, the measured latency density is spread to a larger area, which results in the mean value of the latency to be bigger. This plot shows how the low latency components dominate the measurement when using the prosessing method, and how the new method of using raw sensor data has a more reliable final latency result, which isn't falsified by the incorrect blinking detections.

3.5 Comparison to alternative measurement methods

When comparing this hardware instrumentation based measurement method of see-through latency presented in this thesis to the two alternative methods presented in section 2.3, the cognitive method and the camera feedback method, the one done by using the Buddy system has its advantages. The cognitive method measurement method would require test subjects and someone conducting the test, so it would not be a suitable option for all test cases. If the device is mass-produced, the test process would take too much time and effort compared to an automated one. For research and development purposes, it could be suitable, but again the effort taken by the measurement comes to question. If measurements are repeated multiple times, it would be less of an effort to simply use a machine to do the testing instead of a human. In addition to this, the assumption that human response time remains at constant, might not always be true, which could lead to

falsified results. Lastly, the fact that at least the method described in section 2.3 can't be used for other than video see-through HMDs, makes this method unsuitable for many test cases.

With the camera feedback method, the measurement can be done automatically and without any additional hardware. With this method, the measurement precision was in less than a millisecond range. [22] The measurement itself is a bit more complicated than the hardware instrumentation-based one implemented on the OptoFidelity Buddy. For the see-through latency to be derived from the camera feedback, time-encoding and -decoding needs to be done, and computer vision algorithms need to be applied, possibly lengthening the time it takes to do the measurement due to more complex computations. Also, this kind of latency measurement needs to be integrated into the measured device, and cannot be delivered as a ready-to-use system to a user. This means, that the manufacturers would have to take this latency measurement into account when making the device and implement it as a built-in measurement in the HMD.

Overall, the new see-through latency addition to Buddy gives accurate enough results with the difference between consecutive measurements being less than a millisecond, and provides the user a simple way to measure the latencies of all kinds of MR and AR devices [25]. Especially when the raw data approach can be truly tested on a customer device, is seen if this method gives better test results than the one based on post-processing the signal. The conducted tests indicate that the latency results should be more accurate with the new measurement method based on the raw data approach. The measurement can be almost entirely conducted with existing hardware and software, which makes the measurement more straightforward than other alternatives, like the cognitive and camera feedback methods. Also, the fact that the measurement is fully automated gives this measurement an advantage over many other test methods.

4. CONCLUSIONS

In this thesis, see-through latency measurement methods created at OptoFidelity were documented and analyzed. Their functionality was also compared to some other proposed latency measurement methods. Both see-through latency measurement methods are hardware instrumentation based. That means the measurements are done with an external measurement device, in this case with the OptoFidelity Buddy. The device has two cameras and a color sensor for detecting light from the DUT screen. The first measurement method was based on processing this color sensor signal with a low-pass filter to remove false detections from the signal, and the latency was then calculated from this processed data. In the second approach, raw data was used for the calculations, but the removal of false detections was done by framing the signal and finding the local maximum point inside the frame. If the brightness value of this point was over a certain threshold, it would be marked as a blinking event. From these selected points, the final see-through latency could be calculated.

Two DUTs were used for measurement in the first measurement method. For the first DUT, the Oculus Quest 2, the latency results were accurate, and there were no false detections of blinking events. For the second DUT, Pico Neo 3, there were many false detections, and the final latency results were lower than in reality due to a large amount of false positives at low latencies. Based on this, the first measurement approach isn't universal enough. That is why the second approach of using raw sensor data was created. With this method, the false detections could be removed, and the measurement gave more accurate results than the previous as expected.

For the second approach of using raw sensor data, there was only test data available at the time of writing this thesis. No additional testing other than the one presented in section 3.3 could be done for the second measurement method. From this data can be seen that the measurement performs much better than the first one due to the fact that the removal of false detections isn't dependant on a low-pass filter, which could have unsuitable parameters for some mixed reality devices. In this raw data method, the see-through target blinking interval is known, so the used frames can be selected in a way that there shouldn't be any other activity than the blinking detection. If there is, it is eliminated by choosing the local maximum point inside the frame. This approach gave more accurate results and is less dependent on the used DUT.

REFERENCES

[1] Hancock, P., Kaplan, A., Cruit, J., Endsley, M., Beers, S., Sawyer, B. and Hancock, P. The Effects of Virtual Reality, Augmented Reality, and Mixed Reality as Training Enhancement Methods: A Meta-Analysis. *Human Factors The Journal of the Human Factors and Ergonomics Society* (Feb. 2020), p. 001872082090422. DOI: 10.1177/0018720820904229.

[2] Dubois, E., Gray, P. and Nigay, L. *The Engineering of Mixed Reality Systems*. 1st. Springer Publishing Company, Incorporated, 2009, pp. 33–35. ISBN: 1848827326.

[3] Buker, T., Vincenzi, D. and Deaton, J. 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 (Apr. 2012), pp. 235–49. DOI: 10.1177/0018720811428734.

[4] Gruen, R., Ofek, E., Steed, A., Gal, R., Sinclair, M. and Gonzalez-Franco, M. Measuring System Visual Latency through Cognitive Latency on Video See-Through AR devices. (2020), pp. 791–799. DOI: 10.1109/VR46266.2020.00103.

[5] Azuma, R. T. A Survey of Augmented Reality. *Presence: Teleoper. Virtual Environ.* 6.4 (Aug. 1997), pp. 355–385. ISSN: 1054-7460. DOI: 10.1162/pres.1997.6.4.355.

[6] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. and MacIntyre, B. Recent Advances in Augmented Reality. *IEEE Comput. Graph. Appl.* 21.6 (Nov. 2001), pp. 34–47. ISSN: 0272-1716. DOI: 10.1109/38.963459.

[7] Microsoft. *Online images, Creative Commons*. Nov. 15, 2022.

[8] Tang, Y., Au, K. and Leung, Y. Comprehending products with mixed reality: Geometric relationships and creativity. *International Journal of Engineering Business Management* 10 (Nov. 2018), p. 184797901880959. DOI: 10.1177/1847979018809599.

[9] Chen, L., Day, T. W., Tang, W. and John, N. W. Recent Developments and Future Challenges in Medical Mixed Reality. (2017), pp. 123–135. DOI: 10.1109/ISMAR.2017.29.

[10] Huang, Y., Shakya, S. and Odeleye, T. Comparing the functionality between virtual reality and mixed reality for architecture and construction uses. *Journal of Civil Engineering and Architecture* 13.1 (2019), pp. 409–414.

[11] Rolland, J., Holloway, R. and Fuchs, H. Comparison of optical and video see-through, head-mounted displays. *Proceedings of SPIE - The International Society for Optical Engineering* (Jan. 1994). DOI: 10.1117/12.197322.

[12] *Meta Quest 2: Tähän asti edistynein virtuaalikokemuksemme*. 2022. URL: https://www.meta.com/fi/quest/products/quest-2/ (visited on 10/12/2022).

[13] Carbone, M., Domeneghetti, D., Cutolo, F., D'Amato, R., Cigna, E., Parchi, P., Gesi, M., Morelli, L., Ferrari, M. and Ferrari, V. Can Liquid Lenses Increase Depth of Field in Head Mounted Video See-Through Devices?: *Journal of Imaging* 7 (Aug. 2021), p. 138. DOI: 10.3390/jimaging7080138.

[14] Ballestin, G., Chessa, M. and Solari, F. A Registration Framework for the Comparison of Video and Optical See-Through Devices in Interactive Augmented Reality. *IEEE Access* 9 (2021), pp. 64828–64843. DOI: 10.1109/ACCESS.2021.3075780.

[15] Itoh, Y., Langlotz, T., Sutton, J. and Plopski, A. Towards Indistinguishable Augmented Reality: A Survey on Optical See-through Head-Mounted Displays. *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3453157. URL: https://doi.org/10.1145/3453157.

[16] Yamada, W., Manabe, H., Ikeda, D. and Rekimoto, J. VARiable HMD: Optical See-Through HMD for AR and VR. Oct. 2019, pp. 131–133. ISBN: 978-1-4503-6817-9. DOI: 10.1145/3332167.3356896.

[17] Yan, Z., Du, C. and Zhang, L. Surface Micro-Reflector Array for Augmented Reality Display. *IEEE Photonics Journal* 12.2 (2020), pp. 1–9. DOI: 10.1109/JPHOT.2020.2971622.

[18] Longfei, L., Wenqiang, L. and Zhanjun, Y. Design of retinal scanning display based on holographic waveguide. *2015 International Conference on Optoelectronics and Microelectronics (ICOM)*. 2015, pp. 293–297. DOI: 10.1109/ICoOM.2015.7398825.

[19] *HoloLens 2 Technical Specifications*. 2022. URL: https://www.microsoft.com/en-US/hololens/hardware (visited on 10/14/2022).

[20] *HoloLens 2*. 2022. URL: https://www.microsoft.com/en-us/d/hololens-2/91pnzzznzwcp?activetab=pivot:overviewtab (visited on 10/14/2022).

[21] Aga, H., Ishihara, A., Kawasaki, K., Nishibe, M., Kohara, S., Ohara, T. and Fukuchi, M. 24-2: Latency Compensation for Optical See-Through Head-Mounted with Scanned Display. *SID Symposium Digest of Technical Papers* 50 (June 2019), pp. 330–333. DOI: 10.1002/sdtp.12923.

[22] Sielhorst, T., Sa, W., Khamene, A., Sauer, F. and Navab, N. Measurement of absolute latency for video see through augmented reality. (2007), pp. 215–220. DOI: 10.1109/ISMAR.2007.4538850.

[23] *OptoFidelity BUDDY Performance tester for VR/AR/MR HMDs*. 2022. URL: https://www.optofidelity.com/offering/products/buddy (visited on 10/23/2022).

[24] *OptoFidelity™ BUDDY*. Nov. 1, 2021. URL: https://www.optofidelity.com/hubfs/products/BUDDY-3/OptoFidelity_BUDDY_digital_brochure_A4_english_2021-11.pdf?hsLang=en (visited on 10/23/2022).

[25] OptoFidelity. Company intranet. Nov. 15, 2022.

[26] *numpy.mean*. URL: https://numpy.org/doc/stable/reference/generated/numpy.mean.html (visited on 11/05/2022).

[27] *numpy.std*. URL: https://numpy.org/doc/stable/reference/generated/numpy.mean.html (visited on 11/05/2022).

[28] Keskinen, P. Personal communication. Nov. 15, 2022.