2IMV25 - INTERACTIVE VIRTUAL ENVIRONMENTS

Final Report: Effects of Latency on a 3D Pointing Task

Adam Siemieniuk
0927432
a.i.siemieniuk@student.tue.nl

Roles:
Background Literature Reader
Experiment Designer

Lloyd Lo-Wong
0962399
l.e.lo-wong@student.tue.nl

Roles:
Data Analyzer
Background Literature Reader

Zirui Feng
0947135
z.feng@student.tue.nl

Roles:
Experiment Designer
Implementer

Xin Zhao
1329553
x.zhao1@student.tue.nl

Roles:
Implementer
Data Analyzer

November 15, 2021

***Abstract*—In this report, we document the relevant research, experiment design, implementation overview and result analysis of the "Effects of Latency on a 3D Pointing Task" project. It is a virtual reality system designed to test and analyze the impact of latency on user operations in the virtual environment. In accordance to our research, we selected two types of latency: controller latency and head-mounted display latency. Through the experimental results and user feedback, we confirm that both latency have observable impact on user operations. Meanwhile, the impact grows significantly with the magnitude of the latency. Moreover, the relation between the latency and performance seems to be non-linear. The probable reason is that we have not been able to completely eliminate the influence of other factors than latency on the experimental results except latency.**

I. Introduction

VIRTUAL reality (VR) often aims to create a realistic experience in which users can fully immerse themselves in an artificial world [1]. The level of immersion that can be achieved is influenced by various factors ranging from photorealistic graphical depictions of the environment itself to the ability to interpret user inputs and create believable outcomes from those actions [2]. One particular aspect of VR that has been explored by many researchers [3][4][5][6][7] is the latency of a VR system. Ware *et al.* [3] and Jota *et al.* [8] demonstrated that latency can negatively impacts user experiences and decreases user performance [9].

With constant technological improvements, VR market share keeps rapidly growing as VR increases in popularity [10]. Since head-mounted displays (HMDs), which are considerably different to fish tank VR or CAVE, currently dominate the market, additional investigation on this type of technology is required to provide better insight into the effects that can occur due to their usage.

II. Project Definition

A. Problem statement

The increase in commercially available products such as Oculus Rift [11], HTC Vive [12] or SteamVR [13] popularised HMDs as the way to experience VR. HMDs allow for better user immersion as they can depict egocentric, first-person view thanks to which user can control and experience an artificial world through a virtual avatar.

Those commercial devices utilize input controllers which contain sticks and button similarly to standard video game controllers. Furthermore, the controllers provide six degrees of freedom (6DoF) information about user hands. Additionally, the HMDs also track user head with at least 6DoF [14]. Collection, transfer and processing of all these inputs as well as external factors, such as wireless communication or temporary unavailability of computational resources, introduce delays between physical user actions and their reflection in VR. As shown in other publications [3][5][9], those delays play important role in creating immersive VR experience and can highly impact usability of the system. Therefore, this work aims to further investigate the effects of latency and its impact on user performance.

B. Research questions

In order to investigate the impacts of latency on user performance, the main study objective and multiple sub-questions are defined.

1) Main research question: In this work, the following question is answered:

- *How does latency affect the performance of 3D pointing tasks?*

2) Sub-questions: In order to find an answer to the research question, the following sub-questions are addressed:

- *Does higher latency increase the time it takes to perform a pointing task?*
- *Is relation between latency and performance linear?*
- *Do various types of latency affect performance differently?*

III. Previous Works

A. Difficulty index

To assess the effect of latency on the performance, a way to determine the difficulty of the task has to be established. For pointing tasks, a well established model for difficulty of a task is included in Fitts' law. It uses the following index of difficulty (ID) for one-dimensional pointing tasks:

$$ID_{min} = \log_2 \left(\frac{D}{W} + 1 \right) \tag{1}$$

where D represents the distance to the center of the target and W represents the width of the target.

Though, since Fitts' law is a model used for one-dimensional pointing tasks, it is quite inaccurate and better fitting models have been found which further expand on this equation by taking into account more dimensionality [15][16][17]. For simplicity, the angle between start and target position into account as mentioned in [15] and [16] is not taken. Instead, the the following measure of the index of difficulty for two-dimensional point tasks proposed by Accot *et al.* [17] is used:

$$ID_{acc} = \log_2 \left(\sqrt{\left(\frac{D}{W}\right)^2 + \alpha \left(\frac{D}{H}\right)^2} + 1 \right) \tag{2}$$

where H represents the height of the target and α is a scalar which varies approximately in the range $[1/7, 1/3]$. This formula has been expanded by Grossman *et al.* [15] to 3D. The 3D version of Accot's difficulty index is as following:

$$\begin{aligned} difficulty &= \sqrt{\left(\frac{D}{W}\right)^2 + \alpha \left(\frac{D}{H}\right)^2 + \beta \left(\frac{D}{Z}\right)^2} \\ ID_{acc3D} &= \log_2 \left(difficulty + 1 \right) \end{aligned} \tag{3}$$

where Z is the depth of the target and β is another scalar. Since our target is spherical, its width, height, and depth are equal and it can be simplified to the following equation which is the same as Equation 1:

$$\begin{aligned} ID_{final} &= \log_2\left(\sqrt{\left(\frac{D}{W}\right)^2} + 1\right) \\ &= \log_2\left(\frac{D}{W} + 1\right) \end{aligned} \tag{4}$$

Therefore, ID_{min} is chosen as the difficulty index model.

B. Movement time estimation

A common estimation for the movement time (MT) of the pointing task is as following:

$$MT = a + b \cdot ID \tag{5}$$

where a and b are scalars and are dependent on the input device. However, since latency has been found to be an influencing factor for the movement time of a pointing task as mentioned in [8][9][18], it has to be accounted for in the equation. MacKenzie *et al.* [9] accounted for latency by multiplying the index of difficulty by the latency and adding this to Equation 5 resulting in the following equation:

$$MT = a + (b + c) \cdot ID \tag{6}$$

where c represents the latency.

Combining Equations 4 and 6 gives the following estimation which models the movement time of the pointing task in three dimensions for spherical objects accounting for latency:

$$MT = a + (b + c)\log_2\left(\frac{D}{W} + 1\right) \tag{7}$$

C. Latency types

As VR systems consist of multiple components, several types of latency can be defined. A few different types are more closely examined to identify the ones with the largest potential to have impact on performance.

1) Controller latency: The controller latency can be defined as end-to-end delay between an input being registered by an HMD controller, e.g. one operated by hand, and the registered action taking effect in the VR world. For example, such latency could be observed as a time difference between moving a controller in some direction and the same movement being repeated in VR after some time period.

Multiple studies [3][6][8][19] show that this type of latency has significant and negative impact on performance. In particular, Perroud *et al.* [6] report that latency of $220ms$ decreases performance by 26%.

In terms of input latency analysis, Ware *et al.* [3] successfully modelled the latency impact using a linear model. However, Perroud *et al.* [6] have observed that the performance degradation has also non-linear components. Furthermore, as mentioned by Ellis *et al.* [20], a "move and wait" strategy starts being used for latencies above $300ms$. Additional, non-linear effects can also be expected to be present as in a 3D environment in which each dimension can affect movement differently [15]. Thus, the influence of latency can be expected to be non-linear.

2) Head tracking latency: The head tracking latency, also often referred to as display latency, can be defined as end-to-end delay between pose of an HMD being registered and a corresponding response being executed. For example, such latency would be non-zero when there is a time delay between turning HMD in a some direction and a change in the viewing direction being rendered by the HMD.

The impacts of display latency have been investigated to lesser extent compared to the input latency. Mania *et al.* [21] have shown that the just noticeable difference for VR head tracking latency is as low as $15ms$ regardless of scene complexity. Wilson [22] and Jerald [23] report that the display latency with magnitude of tens of millisecond already degrades performance and can cause motion sickness, such as oscillopsia, in the VR users.

3) Jitter: Beside considering the absolute magnitude of latency, jitter describes the variation in the latency magnitude. Such dynamically changing latency can further amplify its effects on performance as in the case of display latency [22]. However, Teather *et al.* [24] show that latency has much stronger effect as long as jitter has comparably small magnitude. Furthermore, jitter can be potentially minimized or even eliminated by, for example, filtering. A drawback of such countermeasure is that the total latency increases.

After taking into account the impacts of the various latency types described above, the *controller* and the *head tracking* latencies are selected for the experiment. The former is a well established source of performance degradation. The latter has also been shown to negatively affect VR users but overall requires additional research to fully determine its effects. The jitter is not going to be further investigated due to its relatively lower significance and the possibility to further minimize jitter at the cost of higher latency.

IV. Experiment

A classic pointing task [25] extended to 3D environment [15][16][26] is chosen in order to investigate the impact of latency on performance in virtual reality.

A. Design

In this experiment, a participant is positioned in the virtual experiment area with their controller being represented by a sphere that enables interaction with the VR world. The size of the pointer is minimized to make the task more difficult. However, the size is selected such that the sphere can still be easily perceived at an arms length. The pointer movement is directly tied to the participant hand motion captured by the input controller. The range of motions of the pointer is not restricted in any direction.

The experiment consists of multiple trials. Each trial requires the participant to move the pointer as quickly as possible between two points, defined as *starting point* and *target point*, present in the virtual space. These points are represented by spheres. This particular shape has been selected due to its symmetry and rotation invariance which aim to minimize potential influences on the measured performance [27]. The difficulty of each trial is controlled by varying the following three parameters:

- *target size* - defined by the radius of the sphere
- *controller latency* - time delay between participant hand movement (controller input) and the action being reflected in the VR world
- *head tracking latency* - time delay between participant head movement (HMD pose tracking) and movement of the camera in the VR world

Each of these parameters has 3 possible values which gives 27 combinations in total.

The starting point is placed slightly in front of the participant to the right of their starting position within arm's reach. This placement aims to provide easy access and to avoids obscuring target positions. Additionally, this position tries to provoke the participant to move their head, e.g. back and forth between the starting and the target points, such that the head tracking latency is also experienced in each trial. Throughout the experiment, the starting point has fixed size and its location does not change.

The target points can spawn within a cube-shaped volume in front of the participant. The size of the volume is selected such that it fits within field of vision of the HMD. The cube has 27 evenly distributed points in a $3 \times 3 \times 3$ grid. Their positions define where the target points can spawn. The volume is further divided into 3 vertical layers facing the participant. Each layer containing 9 points. From the 27 combinations of the experiment parameters, each combination is used exactly once in each of the three layers. This gives a total of 81 targets in a single run of the experiment.

Teather *et al.* [28] have shown that while participant appreciate presence of extra visual aids, such as pedestals beneath target objects, these cues do not significantly impact overall performance. Thus, no additional visual cues are placed in the environment to reduce the implementation complexity.

B. Apparatus

The experiment is conducted using an HTC Vive [12] HMD device that provides binocular display and head tracking. Additionally, one of the controllers operated by hand is used. The controller provides information about its position and orientation using 6DoF sensor. This allows to translate real world movement of the controller into equivalent motion of the pointer in the virtual world.

C. Participants

Eight male participants from TU/e, both students and employees, have been asked to perform the experiment. All participants are right-handed and have had various levels of familiarity with HMDs ranging from little experience to frequent users.

D. Procedure

In the beginning, a participant is given an HMD and a set of controllers. After the devices are equipped, the experiment application is started. Then the participant receives short explanation about the VR world and is instructed to move the pointer between the starting point and a target point as quickly as possible.

At the beginning, 3 training trials are performed. Their aim is to allow the participant to familiarize themselves with the hardware and the VR environment. No delays are introduced during these trials. Upon completion, the experiment begins.

For each trial, a random layer is selected. Then one of the 27 possible combinations is selected and one of 9 possible target positions of that layer is selected. All of this is done randomly, however, the same combination cannot be used twice in the same layer. Moreover, each position in a layer must have exactly three combinations assigned to it. Immediately upon completion of a trial, another one starts and the experiment continues until each of 81 targets has been hit. The order of the trials is randomized across all participants.

Based on the initial tests conducted on a prototype implementation, a conservative assumption about completion time is made. The estimated upper bound of 10 seconds per trial with 81 possible targets should ensure that the completion time remains within 15 minutes. This maximum experiment duration is selected in order to limit strain on participants.

E. Data analysis

During the experiments, the time it takes to perform each trial for each participant is logged. Each trial has the width of the target, distance from start object to the target object, and latency of the equipment logged as well. Using this data the effect of latency on the performance as well as the effect of width of the target and distance to the target on the performance is analyzed as well as fitted to a model known as Fitts' law [25].

V. Implementation

The main application development tool during the project is Unity [29]. The final version was implemented with the Unity version 2018.3.14f1. Team members were required to use the same specific unity version during the software development process to avoid potential incompatibility issues during testing phase. Meanwhile, Unity scripts that allow to implement specific system behaviors and functionalities are written with the C# language. Therefore, Visual Studio is used to edit and modify the scripts.

In order to test the application without HTC Vive HMD device during the implementation process, a software simulation has been utilized. The simulation solution consists of Riftcat [30] together with Steam VR[13]. Riftcat allows to emulate an HTC Vive headset with a use of smart phone. As the Riftcat are cooperatively running with Steam VR, the software imports the SteamVR Plugin [31] package from Unity *Assert Store*. With help of these tools, the experiment implementation can be tested using a basic PC and a mobile phone as shown in Figure 1.

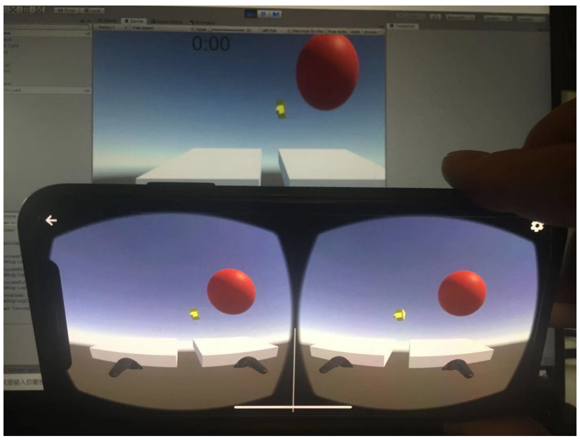


Fig. 1: Testing with Riftcat simulation

In the program, the user operates an HMD controller. It allows to manipulate a yellow sphere which serves as a pointer in the virtual environment. The red sphere represents the starting point, and the user needs to control the yellow sphere to touch the red sphere. In the simulated system, the overall position (including the head and controller) of a person can be manipulated via ctrl for squat, space for high jump, and the arrow keys on the keyboard for walking. The relative position of controller to the head device could be manipulated with combination of keys 1/ 2/3/4 and a mouse wheel. For the real HMD device the head and controller simply move with the exact position of head device and controller device. The green spheres are targets, and they appear one at the time. Every time the controller reaches the starting position, a target appears and the timer begins measuring time. The user needs to touch the target with the controller. The timer switches off as soon as the target is reached. Afterwards, the red starting sphere appears again and the process is repeated until all tasks are completed.

The location of all targets is defined in advance, but the order of appearance is random. In order to minimize the time-consuming error caused by the user in finding the target, there are 8 semitransparent spheres positioned at the corners of the cube. This allows users to discover targets as soon as they appear. In addition, all targets appear in front of the starting position for the same purpose. Then, the user only needs to move the controller in one direction or move forward a short distance. This way users do not lose their way when looking for a target which consumes a lot of time and has large impact on the results of the experiment. The general overview of the program is shown in Figure 2 and Figure 3.

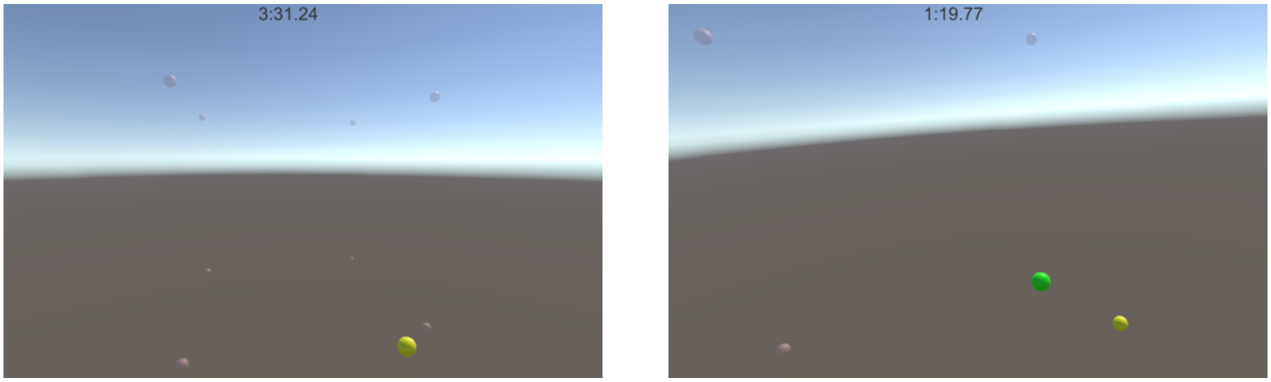
There are two types of latency in the application. The first is the controller latency. It means the user operation of the controller is executed after the default latency. The idea is to store all the user operation tasks in a queue in order. The program continuously checks whether the time difference between the occurrence time of the first task in the queue and the current time is greater than or equal to the latency value. If the result is true, then the program executes the user's operation. Therefore, during the test procedure, it can be clearly observed that the controller in the virtual environment does not move or turn immediately after the user manipulations, and the difference depends on the set latency time.

The second latency is the display latency. The difference is that the program immediately performs the user manipulation but does not immediately display the results to the user. In order to achieve this effect, two different cameras are set in the Unity program. The first camera stores the current view result and the second camera stores the preview result when the user operation is executed. When the user moves the controller, the image of the first camera is still displayed until the set latency time is over. Then the image of the first camera is merged into the image of the second camera. Then the view of the second camera is shown to the user.

In total, there are three controller latency levels and three HMD latency levels. Each time, one controller latency and one display latency are set before user's controller touches the start position. These two latency are accompanied by the program running until the user touches the target. Then the latency levels are be reset to next combination. This design ensures that the time for the modification of latency is not included in the total timing. Because the result data collected by the experiment is the time consumption by the user to complete each target task. Thus, the latency only needs to appear in this process.

In the application, the entire virtual environment is set to 1.6×2 square meters. The diameter of the start position sphere is 0.1 meters. The diameter of the controller sphere is 0.025 meters. The target diameter sizes are selected among 0.025 meters, 0.05 meters and 0.1 meters randomly. The controller latency levels are 0ms, 200ms and 400ms, and the display latency levels are 0, 100ms and 200ms. The reason for choosing these values is that the display latency has been observed to have greater impact on the user than the controller latency. During the initial tests, the delay would become very noticeable by the users whenever the latency value is greater than 100ms. At the levels above 300ms, some users also reported dizziness and increased fatigue.

Finally, the program outputs all the data into a text file



(a) The yellow sphere represents the controller in the virtual environment.

(b) The green sphere represents the target in the virtual environment.

Fig. 2: The overview of the whole virtual environment.

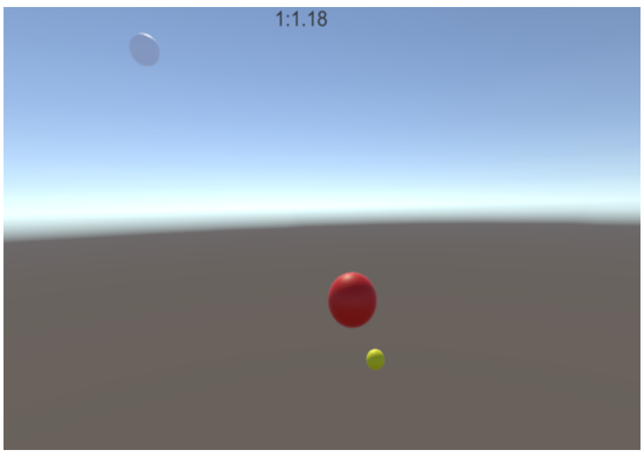
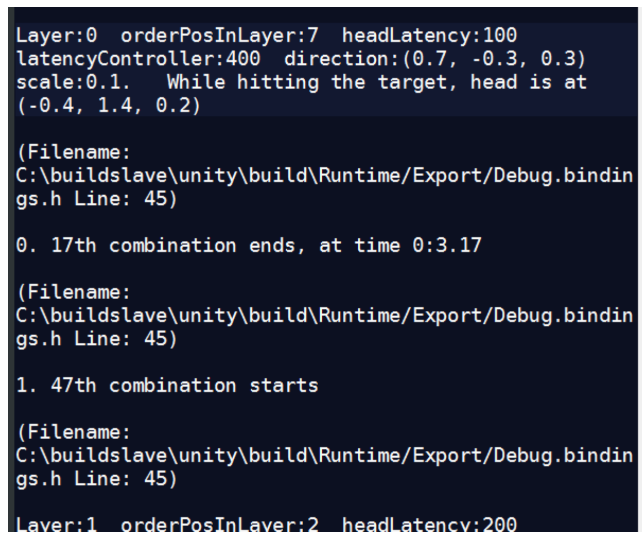


Fig. 3: The red sphere represents the starting position in the virtual environment.



```
Layer:0  orderPosInLayer:7  headLatency:100
latencyController:400  direction:(0.7, -0.3, 0.3)
scale:0.1.   While hitting the target, head is at
(-0.4, 1.4, 0.2)

(Filename:
C:\buildslave\unity\build\Runtime/Export/Debug.bindin
gs.h Line: 45)

0. 17th combination ends, at time 0:3.17

(Filename:
C:\buildslave\unity\build\Runtime/Export/Debug.bindin
gs.h Line: 45)

1. 47th combination starts

(Filename:
C:\buildslave\unity\build\Runtime/Export/Debug.bindin
gs.h Line: 45)

Layer:1  orderPosInLayer:2  headLatency:200
```

Fig. 4: Debug.log output text file example.

by using the *Debug.Log* function in the C# language. The log contains the total time taken to finish all the tasks, individual time consumption, two latency levels, size, distance and location values for each target. The Figure 4 shows the example of the output data.

VI. RESULTS

In this section, the effect of latency on the performance of the pointing task are examined. Furthermore, the data is fit to Fitt's model. Prior to analyzing the data, outliers have been removed for each participant since there were several cases where the inability to find the target influenced movement time severely. For each participant any data point which was outside of the interquartile range of the participant was removed. In total, 6.5% of the data was categorised as outlier and removed from the dataset. The time it took for each participant to hit the targets can be seen in Figure 5a.

A. Movement time

The mean movement time of the participants was 2.115 seconds with a standard error of 55.3 milliseconds prior to removing the outliers. Afterwards, the mean movement time of the participants was 1.883 seconds with a standard error of 28.4 milliseconds.

After repeatedly performing various ANOVA to measure the effect of our independent variables on the movement time, it is found that all the independent variables have an effect on the movement time: width ($F_{2,603}$ = 12.34, p < 0.001); distance ($F_{15,590}$ = 6.39, p < 0.001); controller latency ($F_{2,603}$ = 27.11, p < 0.001); HMD latency ($F_{2,603}$ = 21.37, p < 0.001.) Increase of the width of the object decreased the movement time and increase of the distance, controller latency, or HMD latency increased the movement time. This is in accordance with Fitt's law. There also seems to be an influence of the interaction between the HMD latency and the difficulty index on the movement time ($F_{89,516}$ = 1.340, p < 0.05.) All other interactions

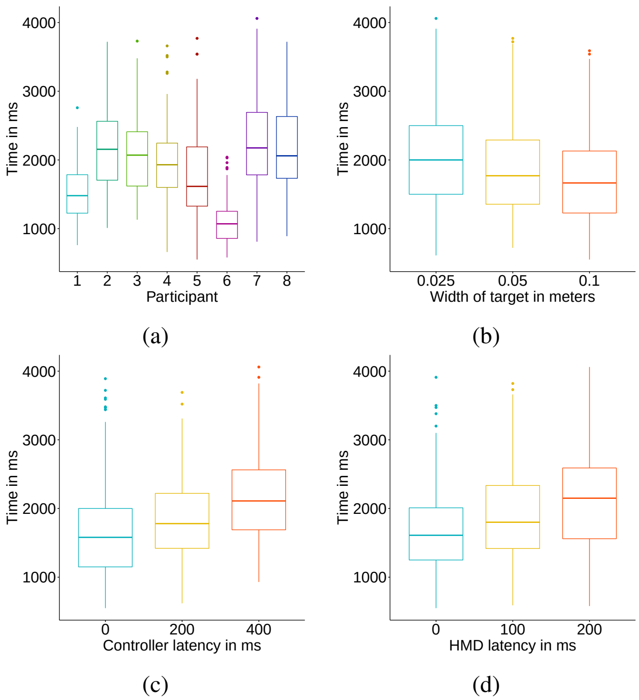


Fig. 5: The effect of various variables on the movement time. (a) The movement time for each participant. (b) The effect of the width of the target on the movement time. (c) The effect of controller latency on the movement time. (d) The effect of HMD latency on the movement time.

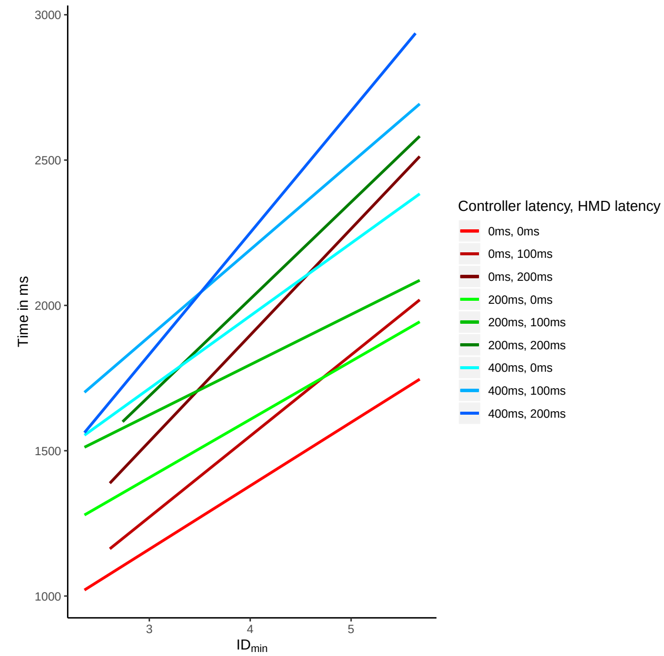


Fig. 6: The correlation between the difficulty index ID_{min} and the movement time, separated by the latency.

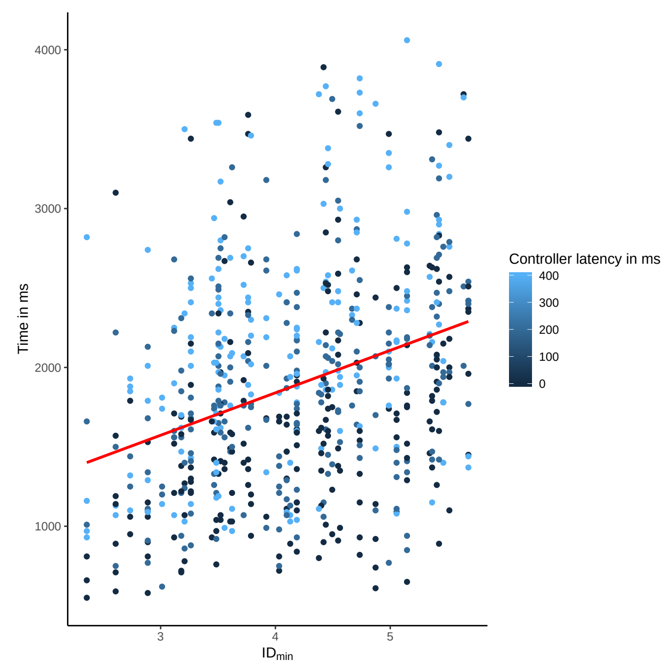


Fig. 7: All the data fitted to Fitt's model, not considering latency.

between the independent variables are of no significance (all had a $p > 0.05$.)

In Figure 5b it can be seen that an increase of the width of the object results in a decrease of the movement time. In contrary, in Figure 5c and Figure 5d it can be seen that an increase of either controller latency or HMD latency certainly increases the movement time. This can furthermore be seen in Figure 6 which shows how the difficulty index increases using different combinations of controller latency and HMD latency.

B. *Fitt's model*

Attempting to fit all our data to the original model given by Fitt which does not take into account latency gives quite a poor fit as can be seen in Figure 7. However, when only considering data with a fixed controller latency and HMD latency, then Fitt's model fits the data a bit better. This can be seen in Figure 8 where the points are much closer to the regression lines.

VII. Discussion

Although the results obtained from the experiment are not fully satisfactory, the sub-research questions can still be partially answered based on the analysis of the resulting data. Meanwhile, a reasonable hypothesis according to the results can be made and possible factors that affect the experimental results can be analyzed.

The research sub-questions and subsequently the main research question are addressed in the sub-sections below.

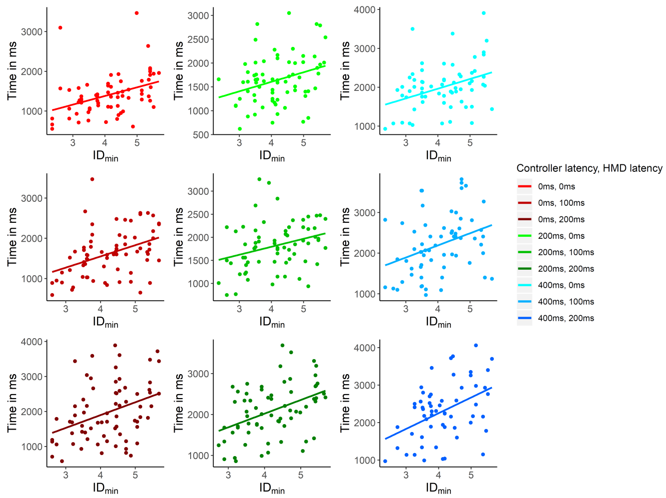


Fig. 8: The data fitted to Fitt's model, separated into groups of latency

A. *Does higher latency increase the time it takes to perform a pointing task?*

There is no doubt that latency can affect the time it takes to perform a pointing task. In the Figure 5 above, as the latency time increases, the average time taken increases significantly regardless of controller latency or HMD latency. Meanwhile, it can also be observed from Figure 6 that the higher latency combination will eventually result in the longest moving time under the same difficulty index.

B. *Is the relation between latency and performance linear?*

By examining Figure 6, it can be seen that performance degradation does not seem to have a linear relationship to the latency levels. For the highest ID, when only HMD latency is varied, the baseline movement time without any delays is equal to $1750ms$. For HMD latencies of $100ms$ and $200ms$, the movement time reaches $2019ms$ and $2513ms$ respectively. Because of this, it can be seen that the performance degrades non-linearly. Similar but less pronounced results can be observed in the cases when only controller latency is applied. The difference between between $200ms$ level and the $400ms$ level is larger than the one between $0ms$ latency and the $200ms$ level. It is also worth noting that a combination of the two delays does not simply stack their effects together. The HMD latency is more dominant between the two delays. However, the controller latency still amplifies the performance degradation.

Overall, the relation between the latency and performance seems to be non-linear which would be consistent with the literature, e.g. [6][20], especially taking into account severity of effect that visual latency can induce at higher levels such as oscillopsia [23]. However, the high spread of data, as shown in Figure 8, highly impact goodness of fit of the performance model which further impact the latency analysis. These issues make it difficult to draw a definitive conclusion.

C. *Do various types of latency affect performance differently?*

The influence of different latency types can be seen in Figures 6 and 8. The controller latency appears to affect the performance in a constant manner across index of difficulty for a given amount of the introduced delay. The slope of the fits representing $200ms$ and $400ms$ controller latency, without any additional HMD delays, is almost parallel to the one of the baseline fit without any delays. The main contribution of the controller latency seems to be an extra constant offset added to the total movement time. This behavior appears to match the nature of this type delay as it is always consistently present throughout the whole movement trajectory.

The constant offset added to the movement time by the HMD latency is lower compared to the controller latency for the same ID. The visual delays have a larger effect on the slope of the fit. As shown in the graphs representing $100ms$ and $200ms$ HMD latency, without any other delays, the degree of performance degradation increases with raising task difficulty. This can be attributed to the fact that this visual delay only affects participants when head movement occurs. As higher ID tasks usually places targets slightly outside of the reach from the starting position or display smaller targets that are harder to notice, the required physical movement of the participants also increases. This in turn increases the amount of exposure to HMD delay compared to when the participant is able to remain in a more stationary position which is mostly the case for the easier targets.

As shown in the gathered experimental data, the actual intensity of the HMD delay, apart from its level, is highly task dependent. Compared to the controller latency that is always consistently present, HMD latency ended up being closely tied to amount of physical movement. This fundamental difference, makes the direct comparison impossible. Based on literature, e.g. [8][20][21], it can be expected that both latency types might not have any lower bound on its influence. However, in regards to the upper bounds, the volatile nature of visual delays can be expected to overwhelm a VR user more quickly than its more constant controller counterpart.

To properly determine the differences between the effects of the two presented latency types, a different experiment design and further research is required.

D. *How does latency affect the performance of 3D pointing tasks?*

As discussed in the previous subquestion sections, there is no doubt that latency affects the performance of a 3D pointing task. A higher value of latency applied to either the controller or the HMD results in an increase of time needed to perform the task. However, we cannot confirm nor deny that the influence of controller latency or HMD latency is linear to the performance. Partly, the reason for this is that in our experiment there are too few levels of latencies for both the controller and the HMD. Furthermore, it would be beneficial to have separately fine-tuned experiment design that can better isolate unaccounted for non-linearities and external factors, such as difficulties in finding targets.

Another reason is that our data has a poor fit to Fitt's model which raises questions on the validity of the data. There are several causes for this poor fit. Firstly, there seems to be a significant difference in time taken between the participants as can be seen in Figure 5a. This is most likely due to the difference of experience of the participants with virtual reality. Secondly, due to our experiment design, participants had to search for the target object after hitting the start object. This introduced another factor which influenced the performance of the task, namely, the ability to find a target in 3D space. This certainly influenced the performance of the task as it could be seen in the data. The inability to find the target was responsible for most if not all of the outliers which were removed from the dataset.

VIII. Conclusion

In this work, the effects of latency on a 3D pointing task in virtual reality have been examined. This was done by conducting an experiment which measured the performance of the participants whilst using a controller and wearing a head-mounted display where both devices were subject to delays. We found that both the latency of the controller and the HMD were factors which influenced the performance of the task. We were unable to find a linear relation between latency and performance due to the experimental design logic and inconsistent result data. Better experiment design and further research are required to fully determine the difference in the influence of the two presented latency types.

References

[1] G. C. Burdea and P. Coiffet, *Virtual reality technology*. John Wiley & Sons, 2003.

[2] H. Limbasiya, "Sense simulation in virtual reality to increase: Immersion, presence, and interactions," University of Dublin, Tech. Rep., 2018.

[3] C. Ware and R. Balakrishnan, "Target acquisition in fish tank VR: The effects of lag and frame rate," in *Proceedings of Graphics Interface '94*, ser. GI '94. Toronto, Ontario, Canada: Canadian Human-Computer Communications Society, 1994, pp. 1–7. [Online]. Available: http://graphicsinterface.org/wp-content/uploads/gi1994-1.pdf

[4] R. Teather, R. Allison, and W. Stuerzlinger, "Evaluating visual/motor co-location in fish-tank virtual reality," 10 2009, pp. 624 – 629.

[5] M. Meehan, S. Razzaque, M. Whitton, and F. Brooks, Jr, "Effect of latency on presence in stressful virtual environments." vol. 2003, 04 2003, pp. 141 – 148.

[6] B. PERROUD, S. Régnier, A. Kemeny, and F. Merienne, "Effects of Latency on Aiming Performance for CAVE-like Immersive Virtual Reality Systems and Driving Simulators," in *Driving Simulation Conference 2018 Europe VR*, Antibes, France, Sep. 2018, pp. 147–150. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01939793

[7] C. Lee, S. Bonebrake, D. A. Bowman, and T. Höllerer, "The role of latency in the validity of ar simulation," in *2010 IEEE Virtual Reality Conference (VR)*, March 2010, pp. 11–18.

[8] 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," 04 2013, pp. 2291–2300.

[9] I. MacKenzie and C. Ware, "Lag as a determinant of human performance in interactive systems," 01 1993, pp. 488–493.

[10] M. Intelligence, "Virtual reality (vr) market - growth, trends, and forecast (2019 - 2024)," 03 2019.

[11] Facebook Technologies LLC. Oculus Rift. Accessed: 2019-09-17. [Online]. Available: https://www.oculus.com/rift/

[12] HTC Corporation. HTC Vive. Accessed: 2019-09-17. [Online]. Available: https://www.vive.com/us/product/vive-virtual-reality-system/

[13] Valve Corporation. Steam VR. Accessed: 2019-09-17. [Online]. Available: https://store.steampowered.com/steamvr

[14] "A quick guide to degrees of freedom in virtual reality," Feb 2018, accessed: 2019-09-17. [Online]. Available: https://kei-studios.com/quick-guide-degrees-of-freedom-virtual-reality-vr/

[15] T. Grossman and R. Balakrishnan, "Pointing at trivariate targets in 3d environments," 01 2004, pp. 447–454.

[16] Y. Cha and R. Myung, "Extended fitts' law for 3d pointing tasks using 3d target arrangements," *International Journal of Industrial Ergonomics*, vol. 43, p. 350–355, 07 2013.

[17] J. Accot and S. Zhai, "Refining fitts' law models for bivariate pointing," 01 2003, pp. 193–200.

[18] C. Ware and R. Balakrishnan, "Reaching for objects in vr displays: Lag and frame rate," *ACM Transactions on Computer-Human Interaction*, vol. 1, 05 2000.

[19] K. Brady, "Taking fitts slow: The effects of delayed visual feedback on human motor performance and user experience," Ph.D. dissertation, 12 2015.

[20] S. Ellis, M. Young, B. Adelstein, and S. Ehrlich, "Discrimination of changes of latency during voluntary hand movement of virtual objects," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 43, 09 1999.

[21] K. Mania, B. Adelstein, S. Ellis, and M. Hill, "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity," 01 2004, pp. 39–47.

[22] M. L. Wilson, "The effect of varying latency in a head-mounted display on task performance and motion sickness," 2016.

[23] J. Jerald, "Latency compensation for head-mounted virtual reality," 2004.

[24] R. Teather, A. Pavlovych, and W. Stuerzlinger, "Effects of latency and spatial jitter on 2d and 3d pointing," 04 2009, pp. 229 – 230.

[25] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, no. 6, p. 381, 1954.

[26] J. Boritz and K. Booth, "A study of interactive 3d point location in a computer simulated virtual environment," *ACM Symposium on Virtual Reality Software and Technology, Proceedings, VRST*, 02 1999.

[27] T. Stuerzlinger, "Considerations for targets in 3d pointing experiments," 10 2013.

[28] R. Teather and W. Stuerzlinger, "Visual aids in 3d point selection experiments," 10 2014, pp. 127–136.

[29] Unity Technologies. Unity Real-Time Development Platform. Accessed: 2019-09-26. [Online]. Available: https://unity.com/

[30] RiftCat Sp. z o.o. Riftcat Vridge. Accessed: 2019-10-01. [Online]. Available: https://riftcat.com/vridge

[31] Valve. SteamVR Plugin. Accessed: 2019-10-01. [Online]. Available: https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647