# Predicting Avatar Movement in Virtual Reality using Neural Networks

Jakob Fehle
Universität Regensburg
Regensburg, Germany
jakob.fehle@stud.uni-r.de

David Halbhuber
Universität Regensburg
Regensburg, Germany
david.halbhuber@stud.uni-r.de

Jonathan Sasse
Universität Regensburg
Regensburg, Germany
jonathan.sasse@stud.uni-r.de

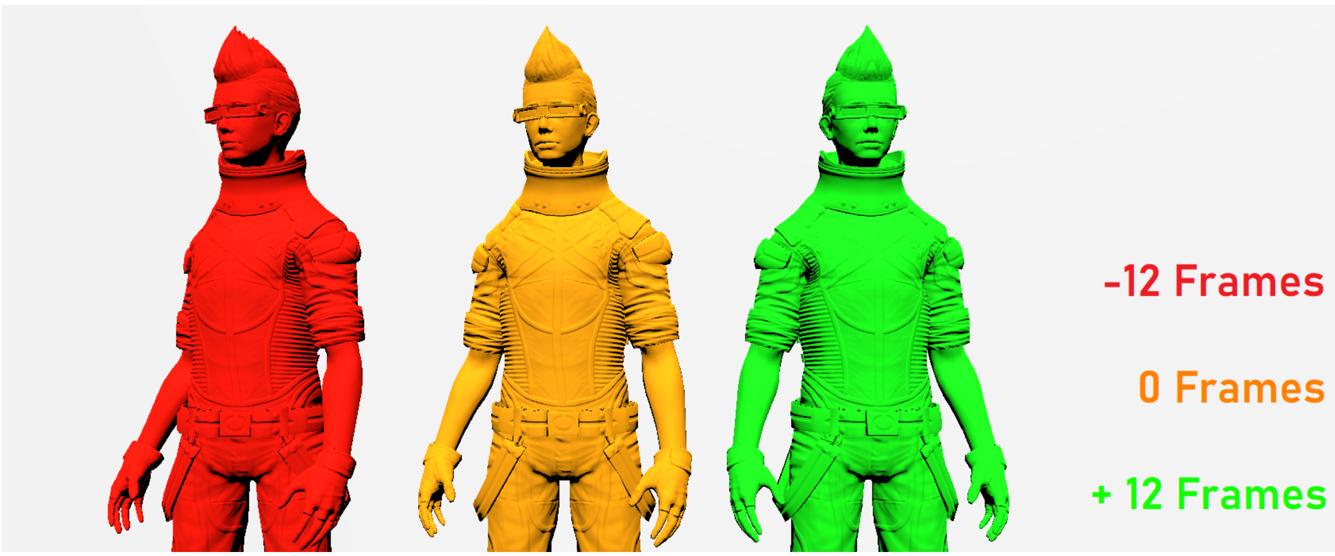


**Figure 1: Showing an avatar whilst moving the upper body and head. The red avatar depicts the -12 frame model. The orange avatar the raw 0 frame model and the green avatar, our predictive 12 frame model. While the green avatar, utilizing the predictive model, has already finished the movement, the red avatar (-12 frame model) has yet to finish the movement. Using the 12 frame model it is possible to achieve a more precise avatar display which leads to a higher location based embodiment.**

## ABSTRACT

In this paper we conducted a study to measure the effects of an avatar movement prediction system in virtual reality on the user's performance and behavior. This prediction compensates the systems latency caused by the motion capturing system and the head mounted device used in the environment. Previous studies have shown that latency has an enormous impact on the user's performance and behavior. To minimize these negative effects, we used a system which is able to reliably determine the system latency and compensate it. To validate our system and measure its effects, we conducted a study with 24 participants. All participants had to complete a performance-oriented task in six different conditions defining the predictive value in frames. One condition was defined as the baseline system latency without the use of a predictive system. Another condition we defined by artificially adding latency, effectively doubling the base system latency. The other four conditions were of a predictive nature - predicating 12 frames, 24 frames, 36 frames and 48 frames in the future. One frame in our setup corresponds to roughly 4.167 milliseconds. In this work we showed that adding additional latency significantly worsen the user's performance. We did not show that reducing, respectively eliminating, latency leads to a better user performance. However, we show, that the use of a predictive model is able to significantly increase the users experienced avatar embodiment illusion. This ultimately can lead to an overall better virtual reality experience for the user.

## CCS CONCEPTS

• **Human-centered computing** → **Interaction design**.

## KEYWORDS

motion capturing, artificial neural networks, movement prediction, virtual reality

**ACM Reference Format:**

Jakob Fehle, David Halbhuber, and Jonathan Sasse. 2020. Predicting Avatar Movement in Virtual Reality using Neural Networks. In *Proceedings of*


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

*ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

# 1 INTRODUCTION

We live in a state of expanded reality where the real and the virtual world blend into each other. Simultaneous interaction with people and machines, such as computers or mobile phones, is nowadays by no means a rarity. Coming from a time where almost no technical artifacts were present in day-to-day activities, the current trend steadily shifts towards a more technical intertwined daily live.

The peak of this intertwining is the so-called Virtual Reality (VR). In VR a person is completely projected into a virtual environment. Users can perceive the virtual world by wearing a head-mounted-display (HMD). By using HMDs, the user is able to see the computer-generated space but is not able to naturally move in and interactive with it. To achieve a full integration of human motion and behavior one needs to use motion capturing devices. State-of-the-art motion capturing systems can be utilized to transfer a user's movement to a computer-generated space. Data received and processed by a motion capturing system can be used for a wide variety of implementations, such as animation creation in movies or controlling an avatar in a VR video game.

Systems combining HMDs and motion capturing systems, have already been widely researched. This is shown exemplary in the work of Chua et al. [2] in which the authors propose and evaluate a method to learn Taijiquan, an ancient Chinese form of material arts, using a motion capture system in combination with an HMD. Although HMD's coupled with motion capturing software are a promising approach for a wide variety of applications, there are some disadvantages and system weaknesses.

From the cameras that are tracking the body movements to the HMD that is responsible for showing those movements to the user there are many technical components involved in the process. Each of these components need time to receive, process and relay the data, which is called latency. A high latency can, for example, induce a user to see his own movements in VR delayed, which in turn leads to a reduction in presence [15, 19] or motion sickness [6]. Furthermore, a high latency value is able to influence a user's overall performance in completing and perceiving tasks [13].

Aim of this work is to compensate for the latency which arises when using HMDs coupled to a motion capturing system. This compensation is achieved by predicting the user's movement through an artificial deep neural network (DNN) [22]. Besides the compensation model we developed three additional models which are able to predict beyond system latency. Finally, we added one model which adds additional latency by holding back the processed movement data in a buffer. To evaluate our models and to measure, quantify and compare effects on users we conducted a study in which users had to carry out a performance-oriented task. We hypothesize that the increase respectively the decrease of the system latency leads to an increased respectively decreased user performance in solving given tasks.

Mainly this paper contributes towards a deeper understanding on how latency and the resolving displacement of the avatar in a VR environment can influence a user's experience and performance. Firstly, we establish an apparatus and define conditions for our test environment. We evaluated and measured the system latency of our test framework. Secondly, we developed a neural network able to predict human movement in VR. We show that a simple dense neural network is able to reliably predict the movement of an avatar in VR. Lastly, combining all our prior efforts we conducted a user study to evaluate the test system and different predictive models. We tested six different conditions; (1) Buffering frames to create additional latency (-12 frame model), (2) unchanged raw data without prediction (0 frame model), (3) predicting 12 frames in the future to eliminate system latency (12 frame model/ zero latency model), (4) predicting 24 frames in the future (24 frame model), (5) predicting 36 frames in the future (36 frame model) and (6) predicting 48 frames in the future (48 frame model). Analyzing our study results we found, that a user's presence feeling does not decrease, to a certain extent, with adding or lessening the experienced latency. On the other hand, we found that performance indeed gets influenced in a negative manner by latency. Most intriguingly we discovered, that our zero-latency system leads to a significant better avatar location perception. This in turn can lead to an overall better virtual reality experience.

# 2 RELATED WORK

This section provides a brief overview of existing research in various adjacent research fields. First of all, we discuss existing possibilities for latency compensation at various degrees of delays and its implication on user performance, as well as their applicability in our scenario. Subsequently, we present papers that deal with the prediction of human movements through neural networks.

## Latency Compensation and Performance Measurement

Wu and Ming demonstrated that latency greatly influences human performance in spatial tasks [31]. The authors developed a system which is able to track and predict the head movement of a human using a HMD. The presented system uses simple linear extrapolation, called *Kalman Filtering* [10], to assume and calculate future head positions. Kalman filtering is based on the assumption that the values to be predicted are describable by a mathematical model, such as a motion equation. A similar assumption is made when using a neural network to predict any value, with the exception that neural networks usually utilize a so-called activation function. The used activation function transforms a former linear regression to a nonlinear problem. Using the Kalman filtering method the authors were able to evidently show that a human can improve task performance by up to 120 % when using latency compensation methods. Since neural networks are capable to coup with more complex problems, using them to predict the movement of the whole body and thus effectively reducing the system latency is a promising approach.

Pavlovych and Stuerzlinger described how latency is able to negatively impact user performance in a target following task [18]. They demonstrated that a latency above a threshold value of approximately 110ms leads to a drastic decrease of tracking performance. In their analysis they showed that the error, and thus the performance, does not increase up to reaching the threshold value of 110ms. Increasing the latency further prevails to an almost linear increase of tracking error. Furthermore, Pavlovych and Stuerzlinger discovered that an increased latency conducts to an increased response delay and thus decreases user performance indirectly. Although the authors only researched mouse-based interaction, the findings are very well usable in our work. Pavlovych and Stuerzlingers work consolidate our hypothesis that latency negatively effects performance.

Waltemate et al examined the impact of latency on perceptual judgments and motor performance in closed-loop interaction in VR [30]. Participants of their study were confronted with their virtual mirror image using a motion capturing system. Measurements included the impact of various levels of delays ranging from 45 ms to 350 ms on motor performance, sense of agency, sense of body ownership and simultaneity perception by means of psychophysical procedures. While motor performance and simultaneity perception were adversely affected by latencies above 75 ms, sense of agency and body ownership required discrepancies higher than 125 ms for that effect even though these aspects were not even at latencies higher than 300 ms fully lost. The authors came to the conclusion that the perception of delays rather depend on the motor task and performance than on the delay itself.

How various delays influence the performance while using interactive systems was researched by MacKenzie and Ware in their publication "Lag as determinant of human performance in interactive systems" [14]. Using four lag conditions, 8.33ms, 25ms, 75ms and 225ms, they conducted a study using the Fitts' law paradigm for measuring the effect in a 2D environment. Participants of their study had to move the cursor by using a computer mouse to a specified target area and pressing a mouse button recording the movement time and error rate. They measured a continuous performance degradation from 8.33ms to 225ms to a total of 63.9% increased movement time and 214% higher error rate. In their conclusion the authors emphasize the importance of considering and taking measures against lags not just in 2D but rather in 3D environments regarding usability and user acceptance.

## Human Movement Prediction through Neural Network

The prediction of motion in VR for various reasons has already been extensively researched. Since HMDs are worn on the head, research focuses mainly on the prediction of head movements. An example is the work of Saad, Caudell and Wunsch in which they analyze the predictability of motion data in a VR environment [21]. They show that the time series data obtained from human head movement are partly deterministic and partly chaotic. This is particularly interesting, because if the data were only chaotic, they could not be predicted. Since the authors showed, however, that the data are also partly deterministic, they can be predicted to a certain degree. The proposed method for checking the predictability of data could also be used in this paper.

Another approach to predicting head motion is shown in the work of LaValle et al [12]. The authors used two different methods to predict the future head position of a user with an HMD. The methods presented are not based on neural networks, but on rather simple mathematical assumptions. In their work, they show that the use of predictive head tracking can drastically reduce latency. For the actual prediction, they used various sensors built into the HMD. Based on the sensor data, especially acceleration data, they built two different prediction models. The first model was based on the assumption that the angular acceleration of the head is constant within a certain time frame. Based on this assumption, the future head position could easily be calculated. The second model assumed that the acceleration of the head was not constant. To calculate a future head position in the second model, the angular acceleration was read from the sensor each time a calculation was required. This resulted in additional processing time. Interestingly, they found that even with the simpler model, at constant angular acceleration, latency could be drastically reduced. A similar approach could be used in this work, assuming that the acceleration of the entire human body is constant in a given time frame. This would greatly reduce the complexity of the problem.

## Summary

Previous work highlights the importance of dealing with delays in 2D and primarily in 3D systems being a major bottleneck in terms of usability and achievable performance. Furthermore, it shows that the compensation of latency in various settings yields promising results. It, as well, indicates that artificial neural networks are well suited for predicting human movement. Through a combination of previous findings, we aimed to develop a system which increases performance in a full body virtual environment. Therefore, we investigated the effect of not just a system without latency but rather a system that calculated and projected future body movements for evaluating the best results regarding performance.

# 3 METHODS

To test our hypothesis, that the perceived latency influences a user's performance, we built a framework to eliminate the latency of our test environment. Core of the framework is an artificial neural network based on a Fully Connected Neural Network (FQNN, DNN), which is responsible for predicting avatar movement in VR. Coupled to a game, which was self-developed in Unity, we conducted a user study in which the test subjects tested different prediction modes.

## Apparatus

To investigate our hypothesis, we developed a test framework containing several components:

(1) *Motion Capturing System*
(2) *Intercepter Client*
(3) *Game*

Using the motion capturing system OptiTrack and the software Motive, body movements of test subjects were recorded at 240 frames per second and could be transmitted over the network. This stream is intercepted in the GUI of our Interceptor client, which modifies the stream data using our pre-trained neural networks. The modified stream gets forwarded to the game which is played by the test subjects. The test subjects perceive the chosen prediction mode in a virtual environment using a non-gender avatar.

To reliably determine the system latency of the used system, we built a latency test framework, which consists of an Arduino microcontroller coupled to a vibration sensor and a photosensitive sensor. The measurement gets initiated by a user dropping an object on to predefined panel. While the object and its fall are tracked by the motion capturing system, the vibration sensor, attached to the panel, is triggered when hit. The Arduino connected to the vibration sensor is informed about the event and writes a timestamp in microseconds, which is then passed to an external computer. A Python script running on this computer receives and sets the timestamp of the vibration as starting value for the latency calculation. In the meantime, the OptiTrack system receives and processes the movement of the object and transfers the information onto a computer. The movement data is being processed and rendered on the HMD using the Unity 3D game engine. Additionally, the Unity application registers the objects collision with the panel. Upon collision Unity starts a function to light up the rendered view which is dark by default. The brightening is registered by the photo sensor detecting the change in light intensity. As soon as the photo sensor is triggered a signal gets passed to the Arduino, which then passes the second timestamp to the external computer. By subtracting the two measured timestamps it is able to reliably determine the system latency. Figure 2 depicts the workflow of the developed latency test framework. With the system latency at hand, we were able to develop a neural network compensating for this very latency.

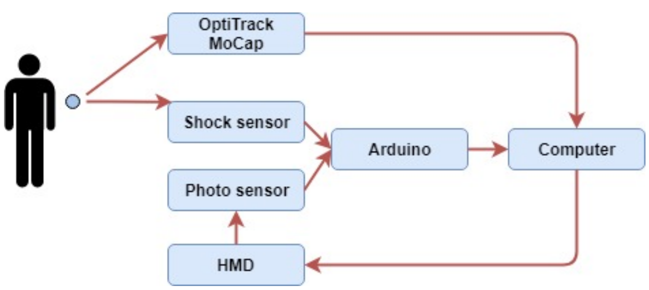


**Figure 2: The workflow of the developed latency test framework. The movement of an object gets tracked by the motion capturing system. The object also triggers a shock sensor attached to an Arduino micro controller and a virtual trigger zone. Triggering this zone commands the computer to brighten the connected HMD. The change of brightness is received by the attached photo sensor. The overall latency is calculated by subtracting the time the photo sensor got triggered by the time the initial shock was registered.**

For the actual movement prediction, we utilized a neural network. The network consists of 87 input neurons that pass data to the first of two hidden layers. The first hidden layer contains 8192 units and is fully connected to the second hidden layer of 8192 units. The second hidden layer was extended by a dropout function to combat overfitting[7, 25]. The dropout rate was deliberately chosen as comparatively low, at 10 percent, to avoid shifting in the underlearning area. Subsequently follows the output layer with built-in ReLu activation function [3].

Since the OptiTrack stream is available in 240Hz and the presented data was also recorded at this frequency, a prediction of 4 frames would correspond to a prediction of about 16 milliseconds. For stochastic optimization we used a variant of the ADAM [11] Optimizer, which is included by TensorFlow. The training process was initiated with a learning rate of 0.001 and a batch size of 512 samples. By using Keras callback functions [1], the learning rate could be dynamically adjusted during the training process. Depending on the validation accuracy, the learning rate was either increased or decreased up to a fixed threshold value. This process is comparable to classical learning rate decay in stochastic gradient descent method implemented in TensorFlow [27], but for our work our implementation proved to be much more effective in dealing with optimization plateaus. The Mean Squared Error (MSE), has successfully proven itself for the loss functions in our test environment. Figure 3 depicts the essential structure of the developed neural network.

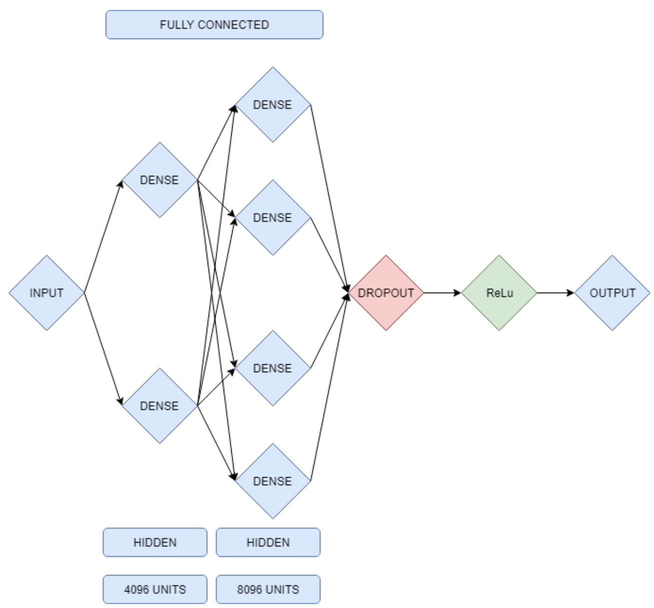


**Figure 3: Shows a schematic representation of the developed network. The two hidden layers are fully connected and are composed of 4096 and 8192 units respectively. In combination with input and output layers, this amounts to a cumulative number of about 9 million trainable parameters.**

The developed game integrates the whole-body movement of the test subject utilizing motion capturing. Through natural movement the player is able to move its avatar in the virtual environment which gets displayed using an HMD. In the game the player is placed in a sort of cage in which the player is able to move freely. The player is not able to leave the cage. The cage has four entrances that allow enemies, in the scenario presented, hostile wasps, to enter the cage. The players objective is to touch these wasps with its hands before they can reach the player's torso. If the player successfully touches the wasp before it touches the players torso the player gets rewarded with one point. On the other hand, if the player does not touch the wasp on time, the wasps earn one point. Figure 9 shows a lateral view of the game, depicting the player, the cage and its surroundings. Figure 5 shows a wasp approaching the player through one of the four gates. The player needs to actively look out for the wasps, since there are four entrances in the cage the wasps are able to spawn and enter the cage behind the player. Lack of awareness of its surroundings or to stationary game play leads to a worse performance. Figure 6 shows a top down view of the play area, depicting the four possible spawn points for the wasps.

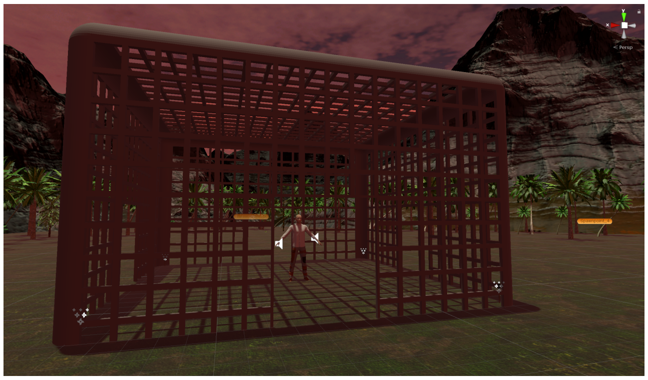

**Figure 4: Shows a lateral view on the game scene, depicting the player, the cage in which the player can move and its surroundings.**

**Figure 5: Shows a wasp approaching the player from inside the cage.**

## Study Design

With the apparatus, consisting of motion capturing system, Interceptor client and game, at hand, a single-factorial within-subject study design was carried out. The dependent variable was set as **game performance** and made up of the players final score. The independent variable was set to **prediction value** which has six levels:

(1) *-12 frames prediction*
(2) *0 frames prediction*
(3) *+12 frames prediction (zero latency model)*
(4) *+24 frames prediction*

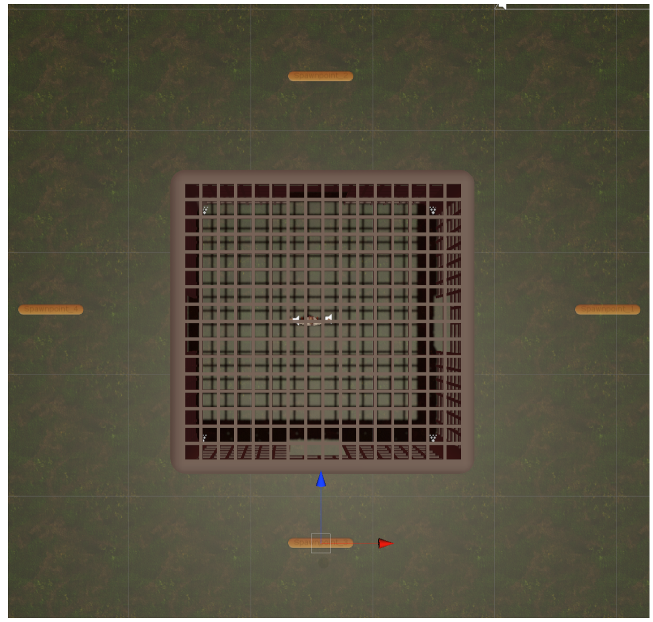

**Figure 6: Shows a top-down view of the cage. Also shows the four possible spawn points of wasps. The spawn points are symmetrically arranged on each side of the cage. The wasps can enter the cage through one of four gates located on the side of the cage.**

(5) *+36 frames prediction*
(6) *+48 frames prediction*

Using our developed latency test framework we were able to estimate the average system latency of a system consisting of the motion capturing environment OptiTrack [17], the game engine Unity [28] and the HMD HTC Vive[8] at 50 ms. This is also the value that gets predicted by the neural network to achieve the zero-latency condition.

## Tasks and Procedure

After giving informed consent, the participants were guided through the study. A total of six conditions were tested:

(1) -12: -50 ms/12 frames prediction
(2) 0: 0 ms/0 frames prediction
(3) 12: 50 ms/12 frames prediction
(4) 24: 100 ms/24 frames prediction
(5) 36: 150 ms/36 frames prediction
(6) 48: 200 ms/48 frames prediction

All participants had to absolve all conditions. The sequence of conditions was balanced via Latin square. The actual task was identic in all conditions. Participants had to play the game for a fixed length of time. They were instructed to score as many points as possible, thus meaning to touch as many wasps as possible before being touched by them. The game was designed in such a way that it was not possible for a test subject to receive the maximum number of points (score). This behavior was implemented to prevent the game becoming a limiting performance factor itself. After each task iteration the participants had to complete several questionnaires. To begin the igroup presence questionnaire[9] (IPQ) was required to be filled out. We use the IPQ to ensure that the feeling of presence perceived by the test subjects does not vary between our conditions. Next probands had to complete the proposed avatar embodiment questionnaire (EQ) of Gonzalez-Franco and Peck [5]. The EQ is used to measure, quantify and reliably distinguish embodiment levels between conditions. Lastly, we asked the test subjects two qualitative questions: (1) if the test subjects experienced motion sickness and (2) asking if they have any further remarks regarding the test system. After answering all questionnaires and questions the probands were tested in the next condition until all conditions were tested.

## Participants

For this purpose, we recruited 24 volunteers (15 m, 9 f) with a mean age of 25,4 years (SD 4,1), which were recorded while participating. A few requirements and criteria for the participants were required in order to capture usable data. Subjects, not meeting those requirements, were excluded from the study and their data deleted, regardless of whether specifications failed prior or during the study. Participants should not be compromised in their movement due to illness, physical disability or high age. Minors did not participate in the study because of the requirement for signing a consent form for using their data which causes an administrative burden. At the start of the study participants had to sign a form of consent to accept the collection of personal data as part of the motion capturing.

# 4 RESULTS

All participants stated that they felt physically and mentally well after the study. The collected data indicates that none of the subjects felt nauseous or dizzy during or after the study. No motion sickness was induced by the use of either of our models (MOS1 + MOS2).

In the evaluation of the data collected through the two questionnaires, a Friedman test could not identify significant differences neither for the categories spatial presence (SP1 - SP5; p= 0.383, F= 1.066), involvement (INV1 - INV4; p= 0.347, F= 1.133) and experienced realism (Real1 - Real4; p= 0.579, F= 0.762) of the igroup presence questionnaire, nor for the categories agency and motor control (AMC1 - AMC3; p= 0.226) and body ownership (BOW1 - BOW4; p= 0.382) of the avatar embodiment questionnaire. Neither is the IPQ subcategory score for the individual categories significant for our different conditions (p= 0.381, F= 1.069). Figure 7 shows the distribution of the averages of the IPQ categories.

EQs category *Location 1,2 and 3 (EQ8, EQ9, EQ10)* revealed near significant results (p = 0.089). Considering this result, we decided to further investigate the corresponding category. *Location EQ9* and *Location EQ10* did not vary significantly (E9: p= 0.414 & E10: p= 0.903). Only for the EQ question *Location 1 (EQ8)* significant differences could be detected using Friedman (p<0.001) and a pairwise Wilcoxon test, the best results were obtained for the 12F model. Figure 8 depicts the distribution of the averages of the EQ categories.

Looking at the performance measurements, a normal distribution of the results over all conditions can be detected with a Shapiro-Wilk test. Considering the mean performance values of the study, the -12 frame model (-12F: 77.5 ± 2.952) and the models that predict further into the future, such as the 36 and 48 frame models (36F: 79.833 ± 3.227 & 48F: 79.5 ± 2.318), perform noticeably worse. The 0F model with native latency performs best in our measurements (0F: 83.833 ± 2.381). Figure 9 presents the mean values of the performance score for each condition.

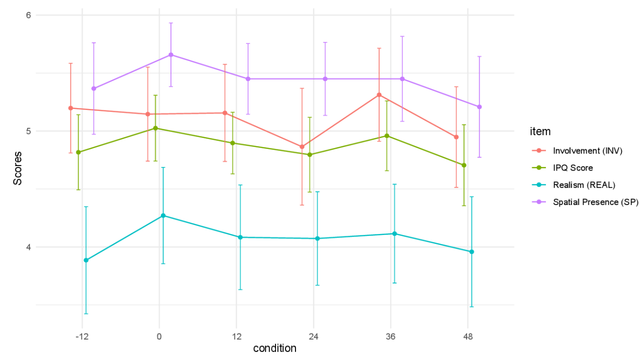


**Figure 7: Distribution of the category averages of the IPQ over all conditions. Error bars show standard deviation.**

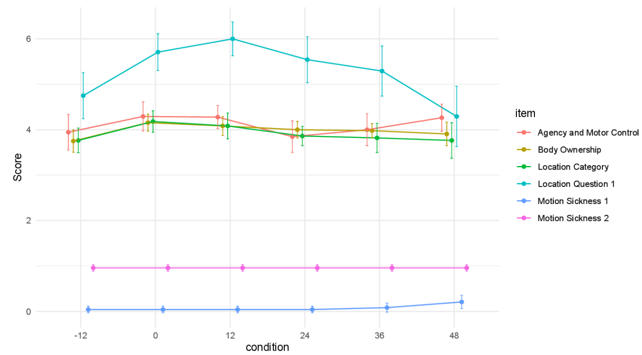


**Figure 8: Distribution of the category averages of the avatar embodiment questionnaire over all conditions including questions about motion sickness. Error bars show standard deviation.**

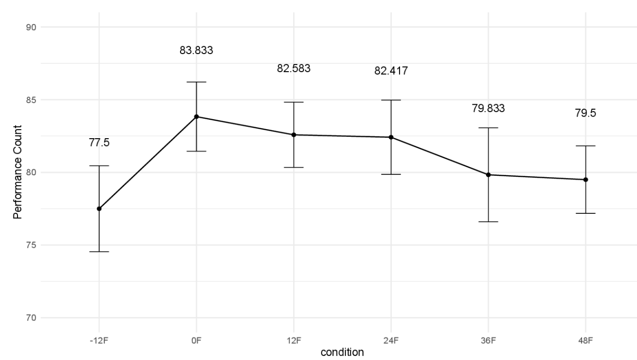


**Figure 9: Mean values of the performance scores for each condition. Error bars show standard deviation.**

A Friedman test indicates a significant disparity in individual conditions of the performance measurements (p: 0.002). The result of a pairwise Wilcoxon test shows significant differences between the prediction models -12F and 0F (p= 0.033) and -12F and 24F (p= 0.028).

## 5 DISCUSSION

Based on our evaluation of the IPQ it can be derived that the usage of any of our models does not negatively nor positively influence the users experienced feeling of presence. Analysis of the EQ shows no significant differences in the categories *Agency and Motor Control*, *Body Ownership*, *Motion Sickness 1* and *Motion Sickness 2*. Especially surprising is the fact that the evaluation of -12F model of the IPQ does not show any significant differences compared to our other tested conditions. This is surprising, because we evidently showed that latency negatively influences performance. We showed that the user's performance was significantly worse using the -12F model compared to the 0F model and the 24F model. Without deteriorating the subjective sense of being in a virtual environment we effectively impaired user performance by introducing additional latency. The behavior shown allows a conclusion to be drawn. Presence and performance seem to be somewhat decoupled in VR. Thus, it could be shown that the performance can deteriorate without damaging the presence. Conversely, it can be assumed that for performance-oriented applications the presence should be of secondary importance. This discovery could be of interest for industrial VR applications, for example in the operation of production machines. However, further research in this direction is necessary to support this hypothesis.

Besides the fact that the user performance deteriorated with higher latency (-12F model) we could not prove our initially established hypothesis that the performance improves with a latency reduction. Although we were able to show that our test subjects perform significantly worse using the -12F model compared to using the 0F or the 24F model, it was not possible to prove a distinct trend or an interaction effect. This could be due to various different factors. Firstly, we assumed that user reaches its peak performance at a zero-system latency model. This assumption could be erroneous, as humans are used to a certain biological latency. This latency is caused by the transmission speed in human nerves [16, 29]. We did not account for the biologic human latency. Instead of eliminating the system latency to zero, we could have lessened the latency to a value which correctly depicts the customary biological latency. Following this assumption, it is possible that real peak performance could be achieved by using a model between the 0F model and the presented 12F model. The 12F model may predict too far in the future to enable a user to perform at an optimum. Additionally, the 12F model predicts per design two milli seconds too far into the future. This is due to the fact, that the smallest unit in our prediction framework is one frame, which corresponds to 4.167ms. This means, that the 12F model, the zero-latency model, predicts 50ms into the future. The measured mean system latency of our system is at 48ms. Using only 11 frames to build the zero-latency model would have resulted in a model predicting only 45.8ms into the future, which would have been not sufficient to eliminate system latency. Assuming that people accustomed to a certain latency due to biological transmission rates, an 11F model might have been the better choice to search for the performance optimum. Additional research in this direction is needed to further investigate which latency leads to peak performance.

Another reason for the lack of performance increase using our predictive models (12F, 24F, 36F and 48F) possibly lays in our used prediction method - the neural network. The artificial neural network is astonishingly accurate (12F = 94.2%, 24F = 91,4% , 36F = 89.5% , 48F = 86.6% ), but cannot generate a 100% accurate prediction. This leads to some movement coordinates being predicted with an error. This error in turn, causes a misrepresentation of the associated part of the body in VR. Since the virtual avatar is updated 240 times per second, this error finally turns into a jittery movement of the corresponding joints. This jitter can be seen as an uncontrolled confounding variable affecting and influencing user performance.

The most intriguing results of this work can be found in the evaluation of the EQ. As already briefly stated in this section, we did not find significant differences in the *Agency and Motor Control*, *Body Ownership*, *Motion Sickness 1* or the *Motion Sickness 2* category using Friedman's test. But using the same method we found significant (p<0.01) deviations in the *Location 1 EQ8* question. This question evaluates upon the question if a participant feels that their body is in the same location as the virtual body. A participant answering these questions with a low score may experience an out-off-body effect or a drifting of the virtual representation of their body [5]. The location of the body heavily affects the embodiment illusion in VR. Comparing the -12F model with the 12F model using a pairwise Wilcoxon test showed a significant better score for the 12F model (p=0.014). Additionally, the 0F model does not produce a significant better score than the -12F model. The same behavior is perceivable comparing the most predictive model. Comparing the 48F model to the 0F model does not show any significant deviations regarding the perceived body location (EQ8), but a comparison between the 48F model and the 12F model reveals a significant better score for the 12F model(p=0.004). This trend is clearly visible in Figure 8 as well. The perceived location, the location category, obtained the highest score while in the 12F condition. Concluding, we found, based on the EQ evaluation, that using our zero-system latency model, the participants had experienced a higher level of avatar embodiment illusion (EQ8). This higher level of embodiment can lead to a increase of subjective presence in VR, which again could translate to a stronger immersion [23, 24]. Additionally, providing the user with a more accurate project of their body in VR enables them to act in more natural manner. Related work shows that increasing the embodiment quality leads to better communication, a better sense of space in VR and even effects cognitive load in a positive way [4, 20, 26]. Ultimately, all these factors lead to an overall better VR experience for the user. Effectively, we were able to correct the avatars body placement for the occurring system latency. Using this correction users perceived a more accurate representation of their own body which led to a higher score in the respective questionnaire category.

## Conclusion and Future Work

We evidently showed that one can achieve a significant superior score in the *Location 1* question of the EQ utilizing a movement prediction system eliminating system latency. This heightened embodiment perception ultimately can lead to an overall improved VR experience for the user. An enhanced user experience is desirable in every application relying on user interaction with any kind of system.

In future work we aim to improve the prediction quality of the neural network. Considering how the network was trained for this work, a better result could be achieved by specializing the network further. For example, one could train a certain task rather than trying to map the whole range of human body movement. Through this specialization the network would be able to more accurately depict frames in the farer future.

Additionally, we plan to further improve the used game. There is reasonable suspicion that users focused more heavily on incoming wasps than on its own body while seeing only its hands if at all. Which leads to a user's reacting presumably rather instinctively than in a controlled, body attentive manner. One the one hand this behavior is highly desired and intended, since we wanted to investigate the effects of our system on the performance. On the other hand, this may lead to the above stated inattentive manner. A better fitting task, respectively game design, has yet to be found and further research in this direction is needed.

## REFERENCES

[1] Francois Chollet. 2019. Keras. https://keras.io/callbacks/. Accessed on 2019-08-12.

[2] Philo Tan Chua, Rebecca Crivella, Bo Daly, Ning Hu, Russ Schaaf, David Ventura, Todd Camill, Jessica Hodgins, and Randy Pausch. 2003. Training for physical tasks in virtual environments: Tai Chi. In *IEEE Virtual Reality, 2003. Proceedings.* IEEE, 87–94.

[3] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics.* 315–323.

[4] Mar Gonzalez-Franco and Jaron Lanier. 2017. Model of illusions and virtual reality. *Frontiers in psychology* 8 (2017), 1125.

[5] Mar Gonzalez-Franco and Tabitha C Peck. 2018. Avatar embodiment. towards a standardized questionnaire. *Frontiers in Robotics and AI* 5 (2018), 74.

[6] Lawrence J. Hettinger and Gary E. Riccio. 1992. Visually Induced Motion Sickness in Virtual Environments. *Presence: Teleoperators and Virtual Environments* 1, 3 (1992), 306–310. https://doi.org/10.1162/pres.1992.1.3.306 arXiv:https://doi.org/10.1162/pres.1992.1.3.306

[7] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[8] HTC. 2015. VIVE Pro. https://www.vive.com/de/product/vive-pro/. Accessed on 2019-05-21.

[9] igroup. [n.d.]. igroup presence questionnaire. http://www.igroup.org/pq/ipq/index.php. Accessed on 2019-05-22.

[10] Rudolph Emil Kalman. 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82, 1 (1960), 35–45.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Steven M LaValle, Anna Yershova, Max Katsev, and Michael Antonov. 2014. Head tracking for the Oculus Rift. In *2014 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 187–194.

[13] Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2122–2131.

[14] I. Scott MacKenzie and Colin Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93).* Association for Computing Machinery, New York, NY, USA, 488–493. https://doi.org/10.1145/169059.169431

[15] Michael Meehan, Sharif Razzaque, Mary C Whitton, and Frederick P Brooks. 2003. Effect of latency on presence in stressful virtual environments. In *IEEE Virtual Reality, 2003. Proceedings.* IEEE, 141–148.

[16] UK Misra and J Kalita. 2019. *Clinical neurophysiology: nerve conduction, electromyography, evoked potentials.* Elsevier Health Sciences.

[17] Naturalpoint. 2015. Optitrack - Motion Capturing System. https://optitrack.com/. Accessed on 2019-05-21.

[18] Andriy Pavlovych and Wolfgang Stuerzlinger. 2011. Target following performance in the presence of latency, jitter, and signal dropouts. In *Proceedings of Graphics Interface 2011.* Canadian Human-Computer Communications Society, 33–40.

[19] Simon Riches, Soha Elghany, Philippa Garety, Mar Rus-Calafell, and Lucia Valmaggia. 2019. Factors Affecting Sense of Presence in a Virtual Reality Social Environment: A Qualitative Study. *Cyberpsychology, Behavior, and Social Networking* (2019).

[20] Brian Ries, Victoria Interrante, Michael Kaeding, and Lee Anderson. 2008. The effect of self-embodiment on distance perception in immersive virtual environments. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology.* 167–170.

[21] E.W. Saad, Thomas Caudell, and Donald Wunsch. 1999. Predictive head tracking for virtual reality, Vol. 6. 3933 – 3936 vol.6. https://doi.org/10.1109/IJCNN.1999.830785

[22] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85 – 117. https://doi.org/10.1016/j.neunet.2014.09.003

[23] Richard Skarbez, Frederick P Brooks, Jr, and Mary C Whitton. 2017. A survey of presence and related concepts. *ACM Computing Surveys (CSUR)* 50, 6 (2017), 1–39.

[24] Mel Slater, Daniel Pérez Marcos, Henrik Ehrsson, and Maria V Sanchez-Vives. 2009. Inducing illusory ownership of a virtual body. *Frontiers in neuroscience* 3 (2009), 29.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[26] Anthony Steed, Ye Pan, Fiona Zisch, and William Steptoe. 2016. The impact of a self-avatar on cognitive load in immersive virtual reality. In *2016 IEEE Virtual Reality (VR).* IEEE, 67–76.

[27] Google Brain Team. 2015. TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD. Accessed on 2019-05-21.

[28] Unity Technologies. 2005. Unity. https://unity.com/. Accessed on 2019-05-21.

[29] P K_ Thomas, TA Sears, and RW Gilliatt. 1959. The range of conduction velocity in normal motor nerve fibres to the small muscles of the hand and foot. *Journal of neurology, neurosurgery, and psychiatry* 22, 3 (1959), 175.

[30] Thomas Waltemate, Irene Senna, Felix Hülsmann, Marieke Rohde, Stefan Kopp, Marc Ernst, and Mario Botsch. 2016. The Impact of Latency on Perceptual Judgments and Motor Performance in Closed-Loop Interaction in Virtual Reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (VRST '16).* Association for Computing Machinery, New York, NY, USA, 27–35. https://doi.org/10.1145/2993369.2993381

[31] Jiann-Rong Wu and Ming Ouhyoung. 2000. On latency compensation and its effects on head-motion trajectories in virtual environments. *The visual computer* 16, 2 (2000), 79–90.

# APPENDIX

| Question | Questionnaire | -12 F | 0 F | 12 F | 24 F | 36 F | 48 F |
|---|---|---|---|---|---|---|---|
| In the computer generated world I had a sense of "being there". | IPQ (G1) | 5.0±0.6 | 6.0±0.4 | 5.8±0.4 | 5.5±0.5 | 5.5±0.4 | 5.4±0.5 |
| Somehow I felt that the virtual world surrounded me. | IPQ (SP1) | 5.3±0.5 | 5.5±0.5 | 5.9±0.3 | 5.9±0.4 | 5.4±0.5 | 5.5±0.5 |
| I felt like I was just perceiving pictures. | IPQ (SP2) | 5.3±0.5 | 5.6±0.4 | 5.5±0.6 | 5.5±0.4 | 5.8±0.5 | 5.2±0.7 |
| I did not feel present in the virtual space. | IPQ (SP3) | 4.9±0.6 | 5.4±0.5 | 4.4±0.7 | 4.9±0.6 | 4.9±0.6 | 4.9±0.7 |
| I had a sense of acting in the virtual space, rather than operating something from outside. | IPQ (SP4) | 5.7±0.5 | 5.7±0.4 | 5.5±0.4 | 5.2±0.5 | 5.5±0.4 | 5.0±0.6 |
| I felt present in the virtual space. | IPQ (SP5) | 5.6±0.5 | 6.0±0.4 | 6.0±0.3 | 5.8±0.4 | 5.7±0.3 | 5.5±0.5 |
| How aware were you of the real world surrounding while navigating in the virtual world. | IPQ (INV1) | 5.0±0.5 | 4.8±0.6 | 5.1±0.5 | 4.6±0.7 | 5.4±0.4 | 4.5±0.6 |
| I was not aware of my real environment. | IPQ (INV2) | 4.7±0.6 | 5.0±0.6 | 4.5±0.7 | 4.8±0.7 | 5.0±0.6 | 4.9±0.6 |
| I still paid attention to the real environment. | IPQ (INV3) | 5.6±0.5 | 5.2±0.6 | 5.1±0.6 | 4.9±0.7 | 5.2±0.6 | 5.2±0.5 |
| I was completely captivated by the virtual world. | IPQ (INV4) | 5.5±0.5 | 5.5±0.4 | 5.9±0.4 | 5.1±0.6 | 5.6±0.4 | 5.2±0.5 |
| How real did the virtual world seem to you? | IPQ (REAL1) | 4.4±0.6 | 4.3±0.6 | 4.6±0.6 | 4.4±0.5 | 4.7±0.5 | 4.5±0.6 |
| How much did your experience in the virtual environment seem consistent with your real world experience? | IPQ (REAL2) | 4.5±0.5 | 5.1±0.5 | 4.8±0.6 | 4.9±0.5 | 4.7±0.5 | 4.3±0.5 |
| How real did the virtual world seem to you? | IPQ (REAL3) | 3.8±0.7 | 4.2±0.6 | 3.8±0.6 | 3.8±0.5 | 3.9±0.6 | 4.0±0.6 |
| The virtual world seemed more realistic than the real world. | IPQ (REAL4) | 2.9±0.6 | 3.4±0.6 | 3.1±0.6 | 3.2±0.5 | 3.2±0.7 | 3.1±0.6 |
| I felt as if the virtual body was my body. | EQ (BOW1) | 4.5±0.6 | 5.8±0.4 | 5.3±0.6 | 4.9±0.5 | 5.1±0.5 | 4.5±0.7 |
| It felt as if the virtual body I saw was someone else. | EQ (BOW2) | 2.9±0.5 | 2.7±0.6 | 3.0±0.7 | 3.2±0.6 | 2.8±0.5 | 3.0±0.6 |
| It seemed as if I might have more than one body. | EQ (BOW3) | 2.6±0.5 | 2.3±0.4 | 2.4±0.5 | 2.3±0.5 | 2.9±0.5 | 3.1±0.6 |
| It felt like I could control the virtual body as if it was my own body. | EQ (BOW4) | 5.0±0.5 | 5.8±0.4 | 5.6±0.5 | 5.5±0.4 | 5.1±0.5 | 5.0±0.5 |
| The movements of the virtual body were caused by my movements. | EQ (AMC1) | 5.0±0.6 | 5.8±0.5 | 6.2±0.4 | 5.2±0.6 | 5.5±0.6 | 5.3±0.6 |
| I felt as if the movements of the virtual body were influencing my own movements. | EQ (AMC2) | 3.6±0.6 | 4.5±0.7 | 4.4±0.7 | 3.7±0.6 | 3.8±0.7 | 4.1±0.6 |
| I felt as if the virtual body was moving by itself. | EQ (AMC3) | 3.2±0.6 | 2.6±0.5 | 2.2±0.4 | 2.7±0.5 | 2.6±0.4 | 3.3±0.6 |
| I felt as if my body was located where I saw the virtual body. | EQ (LOC1) | 4.8±0.5 | 5.7±0.4 | 6.0±0.4 | 5.5±0.5 | 5.3±0.6 | 4.3±0.7 |
| I felt out of my body. | EQ (LOC2) | 2.8±0.5 | 2.9±0.6 | 2.7±0.5 | 2.4±0.3 | 2.5±.04 | 3.2±0.6 |
| I felt as if my real body was drifting toward the virtual body or as if the virtual body was drifting toward my real body. | EQ (LOC3) | 3.8±0.6 | 3.9±0.5 | 3.6±0.6 | 3.7±0.6 | 3.7±0.6 | 3.8±0.6 |
| Did you, at any time during the experiment, feel nauseous or motion sick? ( 0 = No, 1 = Yes) | EQ (MOS1) | 0.0±0.1 | 0.0±0.1 | 0.0±0.1 | 0.0±0.1 | 0.1±0.1 | 0.2±0.1 |
| Do you have any further remarks regarding the system, the qualtiy of the system or the quality of the virutal reality? | EQ (MOS2) | | | | | | |

**Figure 10: Questions asked after each condition. In addition, the abbreviation, the values mean and standard deviation of each question are displayed.**