VR Sickness Prediction for Navigation in Immersive Virtual Environments using a Deep Long Short Term Memory Model

Yuyang Wang* Jean-Rémy Chardonnet† Frédéric Merienne‡

LISPEN EA7515, Arts et Metiers, HESAM, UBFC, Institut Image

ABSTRACT

This paper proposes a new objective metric of visually induced motion sickness (VIMS) in the context of navigation in virtual environments (VEs). Similar to motion sickness in physical environments, VIMS can induce many physiological symptoms such as general discomfort, nausea, disorientation, vomiting, dizziness and fatigue. To improve user satisfaction with VR applications, it is of great significance to develop objective metrics for VIMS that can analyze and estimate the level of VR sickness when a user is exposed to VEs. One of the well-known objective metrics is the postural instability. In this paper, we trained a LSTM model for each participant using a normal-state postural signal captured before the exposure, and if the postural sway signal from post-exposure was sufficiently different from the pre-exposure signal, the model would fail at encoding and decoding the signal properly; the jump in the reconstruction error was called loss and was proposed as the proposed objective measure of simulator sickness. The effectiveness of the proposed metric was analyzed and compared with subjective assessment methods based on the simulator sickness questionnaire (SSQ) in a VR environment, achieving a Pearson correlation coefficient of .89. Finally, we showed that the proposed method had the potential to be deployed within a closed-loop system and get real-time performance to predict VR sickness, opening new insights to develop user-centered and customized VR applications based on physiological feedback.

Index Terms: Human-centered computing—Virtual reality—Walkthrough evaluations; Human-centered computing—User interface design—Interaction devices; Computing methodologies—Machine learning—Machine learning approaches—Neural networks

1 INTRODUCTION

As immersive virtual reality (VR) systems have become available to public consumers, such technology reshapes the way how users interact with virtual environments (VEs). Currently, VR research focuses on many subfields related to human-computer interaction (HCI), e.g., hardware and theory. Hardware engineers try to produce environments that are immersive as much as possible based on new technologies [4, 24], while the theoretical workers tend to design and verify new types of user interfaces and interaction techniques in an immersive virtual environment [2, 18]. With the fast development of technologies coming from video games, new types of devices are frequently released, continually leading to upgraded user interfaces and interaction techniques.

VR headset manufacturers try to provide higher immersion with their products. The counterpart is that users tend to suffer from more cybersickness, getting even worse as the immersed time passes [28]. So far, the lack of efficient methods to eliminate VR sickness in virtual environments still attracts much attention from both academic and industrial communities. In particular, navigation with irregular motion in VEs can induce severe cybersickness according to the well-known sensory conflict theory [8, 30]. In such situation, it would make a difference if at least we could measure and quantify the amount of VR sickness after a user plays with VR systems; and in a latter stage we could be able to design an automatic feedback evaluation system to optimize VR interaction interfaces in real time according to the user's level of VR sickness. This paper proposes to tackle this issue by introducing a deep learning method to predict VR sickness in real time and allow for adaptive interaction in VEs.

IEEE Student Member, e-mail: yuyang.wang@ensam.eu

†*IEEE Member*, e-mail: jean-remy.chardonnet@ensam.eu

‡e-mail: frederic.merienne@ensam.eu

2 RELATED WORK

Crampton categorises the measures for motion sickness into two ways: qualitative and quantitative [23]. Qualitative test scores are based primarily on psychological description or reports of signs and symptoms from experimenters and test subjects. For example, motion sickness questionnaires are commonly used to estimate sickness. However, multiple sources of error are inherent and inevitable as questionnaires heavily rely on subjective evaluations of the indicators being measured; thus the errors tend to undermine the reliability of the measurement procedure or experiment.

In parallel, quantitative assessments through physiological body signals arising from simulator sickness provides experimenters with opportunities to have precise direct comparisons between and within subjects. Another benefit of quantitative measurements from physiological manifestations is the possibility to continuously collect data throughout an experimental session, thereby achieving better accuracy and detail to characterize the progression and effectiveness of sickness symptoms. This also enables to design a real-time estimation system as an efficient sickness indicator.

2.1 Qualitative measurements

In the early 1980s, the motion sickness questionnaire (MSQ) was designed to measure motion sickness arising in different kinds of transport such as cars, buses, ships and airplanes [13]. Kennedy et al. [20] proposed a short version of the MSQ, the now well-used Simulator Sickness Questionnaire (SSQ), considering only 16 items of the MSQ and removing MSQ symptoms that rarely or infrequently occur in simulator exposures, i.e., that are too low to be counted for statistical analysis. Factor analysis methods including principal-factor analysis and normalized varimax rotation were used to identify the coincidence or clusters of symptom items. The SSQ can be divided into three categories: nausea, oculomotor and disorientation. Kim et al. proposed a simplified version of the SSQ, the virtual reality sickness questionnaire, which considers only occulomotor and disorientation [22]. Their questionnaire was shown to be more appropriate in VR studies than the SSQ.

Practical performance and easiness of use have become the most fundamental features of the SSQ when used to evaluate sickness after simulator tests on relevant platforms. Up to now, the SSQ is used in most VR applications. For example, many past studies have been conducted with the SSQ to assess simulator test conditions on educational devices for training [11] and compare the user experience of navigation paradigms [7], or to research on the effect of the

2019 IEEE Conference on Virtual Reality and 3D User Interfaces
23-27 March, Osaka, Japan
978-1-7281-1377-7/19/$31.00 ©2019 IEEE

field of view and image delay of head-up displays on simulator sickness and try to find optimal settings, which is receiving increasing attention [1, 27]. Second, it becomes an effective measurement to quantify the level of simulator sickness and the SSQ generally provides a reference score to be compared with during the development of new indicators as we will see in the following sections.

For example, Padmanaban et al. used ratings on a varied dataset of steroscopic 3D videos based on characterized video features such as the stereoscopic depth, vection and motion velocity to correlate these variables as a function of cybersickness [28]. Meanwhile, the SSQ was introduced to evaluate the level of the corresponding sickness after each experiment. The motion sickness susceptibility questionnaire short-form (MSSQ-Short) [14] was also used to weight the scores among different users for each experiment considering individual differences and to integrate ratings into a single and comprehensive rating.

2.2 Quantitative measurements

Although well-established questionnaires in terms of motion sickness are widely used, there is one evident drawback of such methods if compared to physiological evaluations; for instance, lengthy questionnaires are generally administered after a user has performed an experiment, therefore he/she has to shift attention away from the experiment and on the contrary has to focus on body feelings [10]. Physiological indicators are developed based on physiological signals such as heart rate variability, blood pressure, electrogastrography and galvanic skin reaction [5, 17]. Dennision et al. proposed other indicators that were not much studied in the scope of cybersickness, such as stomach activity, blinking and breathing [10]. From these quantitave measurements, past work proposed strategies to prevent cybersickness from increasing while navigating, typically, the users' electrodermal activity was shown to be an efficient indicator to estimate the level of sickness and was used as a parameter in the navigation control law to adapt navigation in virtual environments, thus significantly reducing cybersickness [29].

Behavioral indicators have received large attention for several years, namely postural instability, as it has been shown to be important indicators of motion sickness [31]. Chardonnet et al. [6] focused on the features of postural sway during immersion and navigation in VR. They collected the postural sway signals in the Left/Right (L/R) and Forward/Backward (F/B) directions and analyzed them in both the time and the frequency domains. They correlated the results with sickness scores obtained from questionnaires. As for the time-domain analysis, the resulting signals of L/R and F/B movements for the marginally stable and the stable states were projected onto an XY plane in order to monitor dynamic changes of the area and shape of the projected signals. The frequency domain analysis revealed two main components: low frequency and high frequency components; the former is usually linked to stable body states in which users do not feel any sickness (voluntary movements) while the latter is more related to marginally stable body states in which sickness arises (involuntary movements).

Unnatural viewing mechanism in virtual environments is also responsible for visual fatigue and VIMS [26]. The observation of depth gaze associated with veridical depth information during exposure in VEs becomes paramount. Wibirama et al. [33] carried out a comprehensive study on VIMS with respect to the SSQ, electrocardiography (ECG) as well as 3D gaze tracking. The occurrence of VIMS was investigated with the SSQ, then the result was utilized for the analysis of user behaviors according to ECG and 3D gaze tracking; they found that participants immersed in 3D visual contents can have less dominant sympathetic nerves activated due to voluntary gaze fixation at one point, which suggests that depth gaze oscillates more frequently when VIMS appears.

2.3 Learning-based indicators

Machine learning is attracting great attention recently as it is supposed to be a powerful tool for data analysis and has already shown many advantages in video games, computer vision, speech recognition [9, 15] or to develop health management systems [12, 21]. Consequently, learning-based algorithms can be extended to predict VIMS if a correlative dataset extracted from physiological or visual signals can be fed into the system.

From the ratings collected through questionnaires, Padmanaban et al. constructed a dataset of stereoscopic 3D videos across a wide spectrum of sickness so that comparative analyses of sickness can be performed in a significant way [28]. A machine learning approach was employed to build a model of the nonlinear relationship between video contents and their corresponding nauseogenicity. However, this method requires artifacts to select features from the videos, which may lead to unavoidable biased results.

Another promising online motion sickness level prediction system is developed for a dynamic vehicle environment based on a learning system [25]; electroencephalography (EEG) signals are used to characterize the physiological changes that occur during the transition of passengers' cognitive states. This information is trained and compared through a self-organizing neural fuzzy inference network (SONFIN) to forecast a sickness level, achieving an overall accuracy of about 82% through experiments. Although this system is built mainly for estimating sickness for passengers in a vehicle, it can also be used to measure cybersickness in virtual reality as the symptoms between motion sickness and cybersickness are quite similar, which has already been demonstrated [19].

In this paper, we concentrate on visually induced motion sickness (VIMS) during navigation in an immersive virtual environment. A set of symptoms may appear on users owing to their susceptibility to VIMS during or after being exposed to dynamic visual displays for a long time. Body postural instability is one of the behavioral phenomena that has already shown effectiveness to capture VIMS [6]. In order to capture user's cybersickness, we propose a new method based on the deep long short term memory (LSTM) network and build an network to characterize the user's level of VR sickness.

The main goal of our work is to estimate and predict the VIMS level based on postural sway signals and ground truth measurements from SSQ scores. Since current approaches to measure VR sickness can only be used in pre-exposure and post-exposure in the VE, we show that the advantage of our method is that it can not only measure the VIMS but also it has the potential to be implemented for real time applications, which enables to get rid of intermittent evaluation methods, on the contrary of [6]. The long-term objective is to extend these results to larger cases and find objective and online measures to quantify simulator sickness with physiological signals. Our objective is a wide one while in this paper we focus on using the postural sway signal. The most important contribution lies in the proposition of a new method to evaluate VR sickness based on an LSTM network which has the potential to be deployed in online applications. The paper is structured as follows: we will first detail the basics of LSTM, then we will show how the LSTM network has been applied to predict VIMS by conducting a user study. We will finally show how our method can be deployed in real-time applications.

3 Principles of VR Sickness Evaluation

3.1 LSTM in deep learning

In this paper, a deep long short term memory (LSTM) network is designed with the aim of detecting and measuring the degree of VIMS. LSTM is one of the methods in the large deep learning family. Deep learning is a method trying to learn representations from source data with successive layers; the term *deep* in *deep learning* is not linked to any kind of deeper understanding of the

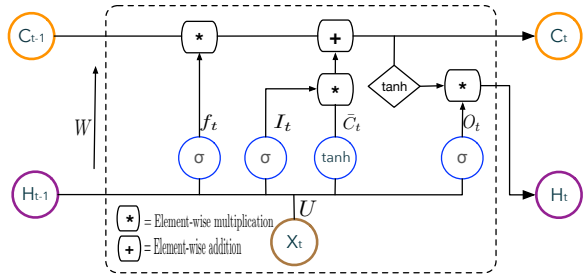

Figure 1: Diagram for an LSTM cell at time step t; W and U are weight vectors for the forget gate (f), the candidate ($\bar{C}$), the input gate (I) and the output gate (O); X_t: input vector, H_{t-1}: previous cell output, C_{t-1}: previous cell memory, H_t: current cell output, C_t: current cell memory; σ: sigmoid activation function.

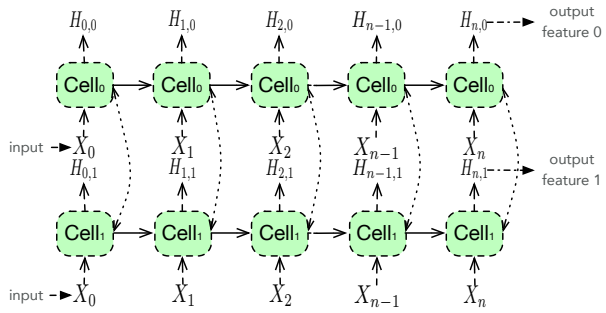

Figure 2: One LSTM layer is composed of two cells that can work in parallel and communicate between each other during the learning process; the input layer is the sequence consisting of a vector with length n; each cell is fed with the same input sequence.

solution achieved by the approach; instead, it is related to the idea of successive representation layers.

Fig. 1 explains the components of one LSTM cell and how it works. The LSTM cell takes inputs from the current step state (X_t), the previous hidden state (H_{t-1}) and the previous memory state (C_{t-1}), and returns outputs including the current hidden state (H_t) and the current memory state (C_t). The internal computational flow is controlled by four different gates which can also be understood as filters: a forget gate (f), a candidate ($\bar{C}$), an input gate (I) and an output gate (O), described in Equation 1 where X_t and H_{t-1} are processed by corresponding weight vectors W and U. To further include history information, current outputs have to join a previous cell memory C_{t-1}, described in Equation 2.

$$\begin{cases} f_t = \sigma(X_t * U_t + H_{t-1} * W_f) \\ \bar{C}_t = \tanh(X_t * U_c + H_{t-1} * W_c) \\ I_t = \sigma(X_t * U_i + H_{t-1} * W_i) \\ O_t = \sigma(X_t * U_o + H_{t-1} * W_o) \end{cases} \tag{1}$$

$$\begin{cases} C_t = f_t * C_{t-1} + I_t * \bar{C}_t \\ H_t = O_t * \tanh(C_t) \end{cases} \tag{2}$$

A single LSTM cell can be extended along the sequence direction in order to process sequential data, which is the original LSTM model with one layer, as shown in Fig. 2. The deep LSTM model, which is a further extension of the original model and that we consider here, contains multiple hidden LSTM layers to learn more accurately the description of the original sequential data. Based on different problem types, the model can output a sequence data or only a single scalar. Especially, when the model learns the features of the input sequence itself and tries to reconstruct such input in the output, the network is called an autoencoder. The so-called features of the sequence are the intermediate and equivalent representations of the original input within a latent space.

Compared to other neural networks like multilayer perceptrons (MLP) and convolutional neural networks (CNN) which have no internal memory and only process each input data independently, LSTM model can perform better in processing long-term dependencies among sequential data with gates including the forget gate, the input gate as well as the output gate, and overcome gradient explosion and vanishing issues at the same time [3]. This motivates our choice for LSTM model to detect and predict cybersickness. Here the LSTM model will be used on postural sway signals to detect cybersickness.

Fig. 3 presents an overview of the proposed method for objectively measuring the VIMS score based on the LSTM model. As shown in this figure, the model consists of an encoder that tries to represent the raw postural sway signal in a latent space, followed by a decoder that tries to reconstruct the original signal from the encoded features. Two stages compose the method: a training stage and an evaluation stage.

3.2 Deep LSTM model for normal sway signals

In the training stage (Fig. 3 left), the LSTM neural networks learns to extract features of the input consecutive signal information by minimizing the loss (mean-square error, MSE) between the original inputs and the reconstructed outputs, thanks to Equation 3:

$$\arg\min_w \mathscr{L} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{X}_i - \hat{f}_w(\mathbf{X}_i)|| \tag{3}$$

where w is used to describe the parameterized model; $\mathbf{X}_i$ is the i^{th} sampling point of the postural sway signal $\mathbf{X}$; N is the length of the signal (number of sampling points); $\hat{f}_w$ is the non-linear function of the neural network that reconstructs the original input with the network. The input signal is a normal sway signal, representing the user's body postural sway when the user is in his/her usual state of fitness (no sickness).

Fig. 4 shows the LSTM model aiming at retrieving the input postural sway signal. The model is built by stacking ten LSTM layers and one fully connected layer (also called dense layer in some literature); the number of layers was decided with the grid-search method where we tried to find the best performance by tuning the model. Deep LSTM layers are used to learn the sequence features. The model includes therefore two main parts that are the encoder and the decoder; it is able to reconstruct the original input with high accuracy. It is worth noting that high accuracy is valid only when the model reconstructs the signal from a similar body state, not to reconstruct every signal from diverse body states with high precision. Each layer is followed by a dropout function to avoid overfitting [32]. Since we want to reconstruct the original input, a dense layer is used to reshape the output feature from the last LSTM layer. This process can be assimilated as a *reshape* function in many programming languages.

3.3 Detection of abnormal sway signals

After training our LSTM neural networks, the VIMS score can be calculated according to the reconstruction errors, as done in [16] (Fig. 3 right). Based on the learned model, the reconstruction error E of a test signal $\mathbf{Y}$ can be written as

$$E = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{Y}_i - \hat{f}_w(\mathbf{Y}_i)|| \tag{4}$$

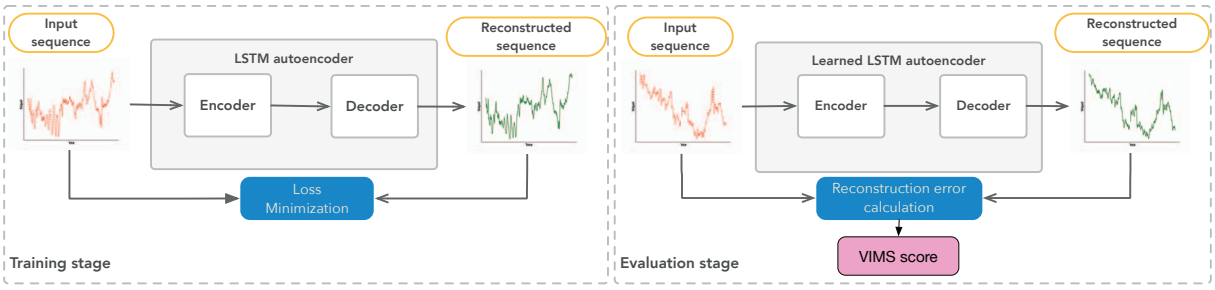

Figure 3: Overview of the proposed method for estimating the VIMS score using an LSTM model.

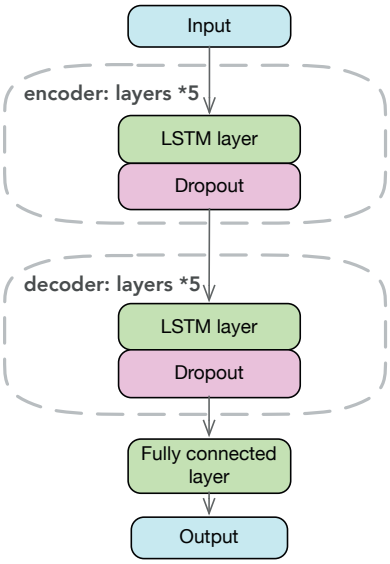

Figure 4: Overview of the proposed method to reconstruct the input signal using LSTM neural networks including five layers for the encoder and five layers for the decoder; dropout is introduced to avoid overfitting.

where $\hat{f}_w$ is the learned LSTM model.

Because the deep network is trained with the postural sway signal obtained before the user is immersed in VEs, the trained model is supposed to capture the features of a person without any feeling of VIMS, and therefore, it can reconstruct the original sway signal. However, when such trained model is used in the evaluation stage in which the postural sway signal is obtained after the user is exposed to VEs, the reconstruction error shall be larger. As a consequence, by evaluating the VIMS score based on the reconstruction loss with a trained model, the level of VR sickness is expected to be estimated and predicted in real time without providing additional information.

A lower reconstruction error means that our model is able to retrieve the signal with more accuracy. In other words, when we train the model with a postural sway signal obtained from a normal state of the person (without any feeling of VR sickness), the encoder extracts the features of such state and the decoder reconstructs the signal accordingly with high accuracy; however, during the evaluation stage, when we feed the network with a postural sway signal obtained from a different body state, the decoder will not be able to reconstruct the input signal with accuracy as the network does not recognize the learned features: the reconstruction error is thus supposed to be large.

4 User Study

We conducted a user study in order to build the datasets to train our deep LSTM model and validate the general method. We carried out an experiment to estimate the level of VR sickness during navigation in a virtual environment through both the postural sway signals and the SSQ. The final results from both measurements were then fitted and correlated to our proposed evaluation method to validate its efficiency.

4.1 Participants

We asked 11 participants including 4 females ($M_{age} = 25.83$, $SD = 4.58$) to take part in the experiment. All participants were recruited through word of mouth. No compensation was given after the experiment. A brief training was provided before the experiment to let the subjects understand their main task in the virtual environment. All participants were requested to fill a pre-exposure questionnaire (Q1) in order to get a general information about their health condition and their background knowledge on computers and virtual reality. According to the results from Q1, all subjects were in normal or corrected-to-normal health conditions, reported no disorders or unusual circumstances with respect to their hearing or balancing, and did not report any severe susceptibility to motion sickness.

4.2 Experimental equipment

To navigate in the virtual environment, an HTC Vive head-mounted device (HMD) was used together with the two wireless handheld controllers provided with the HMD to control navigation. In our experiment, the trigger button and the trackpad of the handheld controller were mainly used to control the speed and the direction of navigation. To move forward, users pressed the trigger button and the movement started with a speed depending on how much the trigger was pressed, while turning was done through a gaze-directed technique in which the moving direction was the gaze direction (here, we assimilated the gaze direction to the head orientation). Users could stop navigation if they released the trigger button or when they arrived at the target position. The HMD was connected to Unity3D via SteamVR and VRTK SDK which provides many built-in APIs for the development of VR user interfaces. With this SDK, interaction between users and the VE could be achieved in real time without any delay.

The postural sway signal of the participants was collected through a Stabilotest balance board from TechnoConcept[1]. It has internal sensors that are able to capture left/right and forward/backward sway signals in real time for 51.2 s with a frequency of 40 Hz, providing 2048 sampling points.

4.3 Experimental procedure

The procedure was designed as follows:

[1]http://www.technoconcept.fr/shop/index.php

Figure 5: Virtual scene where the participants complete the task.

1. All participants were requested to fill questionnaire Q1 before the experiment. Considering that the participants were unfamiliar with the HTC Vive, we gave them a brief training on the tasks including what they were going to do and on how to control navigation with the controllers.

2. In order to measure the difference of user's level of cybersickness before and after the main task, each participant had to fill an SSQ_{pre} (Q2) to get a pre-exposure VIMS score then had to stand on the balance board while looking at a fixed point displayed on the wall to record the corresponding sway signal. Note that though past work has recommended not to administer an SSQ before exposure [34], rather than considering participants to have no sickness at all, we chose to administer the SSQ to get a baseline of participants' actual state.

3. The participants navigated through the VE and completed the given task. In order to provoke VIMS at different levels, each participant had to search and collect the coins scattered in the VE (Fig. 5), and we set the maximal navigation speed to $2\mathrm{m}/\mathrm{s}$ which is slightly higher than the normal walking speed. The participants were urged to explore and pick up a total of 30 coins as fast as possible. All along the experiments, they were in a standing position, which is supposed to favor cybersickness more compared to a sitting position [31]. Note that the motion speed might be less than the maximal speed during task completion, depending on the input from the controller.

4. After the participants finished the navigation task in the VE, they took off the HMD then stood again on the balance board to collect another sway signal. Besides, another SSQ_{post} (Q2) was filled to calculate the ground truth VIMS score after navigation in the VE.

5. The participants were invited to do the navigation task again within the same experimental conditions on another day in order to have more data samples, repeating the procedure between 2 and 4.

The whole procedure of the experiment is given in Fig. 6. The participants were exposed to the VE each time between 3 and 5 minutes. After finishing the whole experiment, we obtained twice four different data for each participant: a pre-exposure postural sway signal and the corresponding SSQ_{pre} score, a post-exposure postural sway signal and the corresponding SSQ_{post} score. Although the participant may become familiar with the environment and its procedure in the second experiment, this would not affect the final result from the participants as the idea was to find the variance of the SSQ score and the corresponding postural sway signal. The LSTM model was then created and implemented using a deep learning framework called Tensorflow[2]. Before the training and evaluation stages, the signal data was pre-processed by centering and scaling between -1.0 and 1.0. The pre-exposure signals were used to train the model so that it can extract the features for the normal state and save them into the LSTM cells. The training stage stopped when the reconstruction error converged to a stable value. Then the post-exposure signals were fed into the model, which led to a jump of loss (strong variation of reconstruction loss) in the same way as described in Sect. 3.3. This jump of loss was then considered to originate from the presence of VR sickness.

5 Performance and Validation of the Proposed Method

To be sure that the sudden variation of reconstruction error between the training stage and the evaluation stage was due to the onset of VR sickness, we correlated it with the level of sickness provided by the sickness scores from the simulator sickness questionnaires. We computed for each experiment the difference between the pre-exposure and the post-exposure scores and we will call it from now on the SSQ score:

$$SSQ = SSQ_{post} - SSQ_{pre} \tag{5}$$

and similarly, the variation of reconstruction error, that we will call from now the loss L, was computed by finding the difference between the average of the last five minimized pre-exposure reconstruction losses and the average of the initial five post-exposure reconstruction losses (we chose to take five values in each reconstruction loss to get a representative loss considering the instabilities and oscillations during the deep learning process):

$$L = \frac{1}{5}\left|\sum_{j=1}^{5} E - \sum_{k=N_L-5}^{N_L} \arg\min_{w} \mathscr{L}\right| \tag{6}$$

where E and $\arg\min_w \mathscr{L}$ were defined in Equation 4 and Equation 3 respectively; N_L is the length of the trained reconstruction error.

Fig. 7 shows the reconstruction losses for one participant provided by the normal state sway signal (pre-exposure, blue curve) and the post-exposure sway signal (green curve). We split the normal state signal into small pieces (one signal with length of 2048 to 32 signals with length of 64) as a pre-processing step to train the model. We can observe the changes of the reconstruction loss during the training stage (80% of the signal pieces) and the evaluation stage; a validation loss (20% of the signal pieces) was also plotted to make sure that there is no overfitting with the dropout method. The procedure starts from epoch 0 (an epoch corresponds to the period of an entire processed dataset) with an initial reconstruction loss of around 0.576 that is due to the random initialization of the deep neural network. The loss then decreases significantly as the epoch passes, then it converges to a stable value. The training stage stops at epoch 1000 as the reconstruction error is stable enough. The model can be deployed subsequently for the evaluation stage where the model is fed with the signal obtained after the user was exposed to the VE. As the model has never been trained to recognize a signal obtained from post-exposure in the VE, the features are impossible to extract and accordingly, the model is unable to reconstruct the signal with high accuracy. Therefore, as expected, we observe a significant jump of the error at around 0.564; this jump and the first values of the post-exposure reconstruction error are the only parts that interest us, as it shows that an abnormal signal is detected, i.e., an equivalent description of VIMS occurs. In this example the loss L is equal to 0.385 while the measured SSQ score is equal to 7.48.

Since all the participants performed the experiment twice, we collected a total of 22 groups of data to be correlated. Table 1 shows

[2]https://www.tensorflow.org

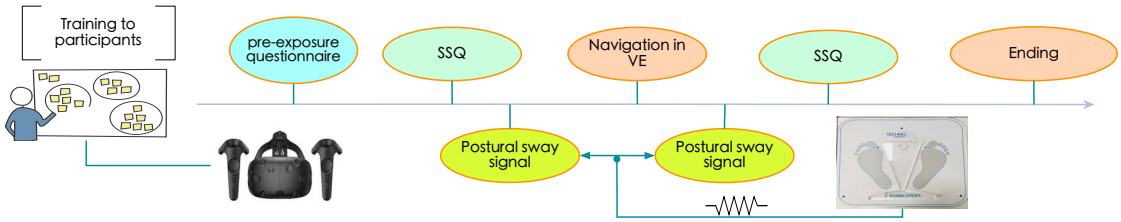

Figure 6: Experimental procedure conducted to obtain the datasets.

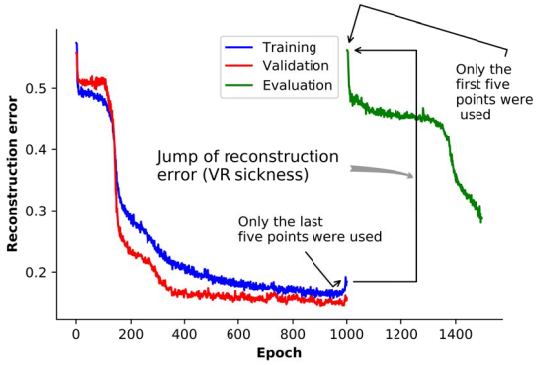

Figure 7: Demonstration of the proposed method to detect VR sickness from the signals of one participant.

Table 1: Correlation matrix

	Loss	SSQ	Nausea	Oculomotor	Disorientation
Loss	1.00	⋱	⋯	⋯	⋯
SSQ	.89	1.00	⋱	⋱	⋮
Nausea	.75	.90	1.00	⋱	⋮
Oculomotor	.77	.87	.70	1.00	⋱
Disorientation	.84	.86	.68	.61	1.00

the correlation matrix that was obtained by computing the Pearson correlation coefficients between all the measured variables, and after decomposing the SSQ score into the three categories (nausea, oculomotor, disorientation). We see that the Pearson correlation coefficient is $r = .89$ between the SSQ score and the loss. The correlation between the loss and nausea is .75, the correlation between the loss and oculomotor is .77, and the correlation between the loss and disorientation is .84. All these correlation values range from .6 to .9, which implies that our new method has enough accuracy to be effective. Fig. 8 shows the correlation obtained between the SSQ scores and the losses. We observe that the loss is scattered along a regression line, which implies that, mirroring the correlation matrix of Table 1, the loss can be an effective method to capture the level of user's VR sickness. However, we also observe that this method becomes deficient when the SSQ score becomes too large, more than 120.

6 Prediction of VIMS with a Closed-Loop System

In the previous section, we separated the experiment into "pre-exposure" and "post-exposure" with two distinct measurements as we wanted to carry out a correlation test with the SSQ to validate the reliability of the model. However, from the results, since we found a strong correlation between the loss and the SSQ score it is also possible to develop an online system with the proposed method without interrupting the experiment. In Fig. 9 we show the architecture of a real-time implementation of the system. The system consists in two steps but only the second step is implemented as a closed loop:

1. The user has to take the HMD on and get immersed in the VE. In order to collect normal physiological signals (no VR sickness felt as at this stage there will no visual vection), she/he just has to stay in place without navigating, avoiding any occurrence of cybersickness. In addition, the signal can also be collected without getting immersed in the VE, which also makes sure that the user produces absolutely normal physiological signals. In either case, the user should not feel VR sickness during this stage so that the model is able to characterize the normal state. After the data acquisition, the LSTM model (also named VIMS detector) can be trained to learn the features of the input signals.

2. The pre-trained model is deployed in the navigation system. Sensors keep tracking the physiological signals of the user and sends the information to the VIMS detector which will calculate online the reconstruction loss. If the reconstruction loss is larger than a pre-defined threshold, then the system detects VR sickness successfully; the system can then alert the user or adapt navigation parameters according to the detector. This step is a closed-loop implementation which shall improve efficiency and user experience significantly.

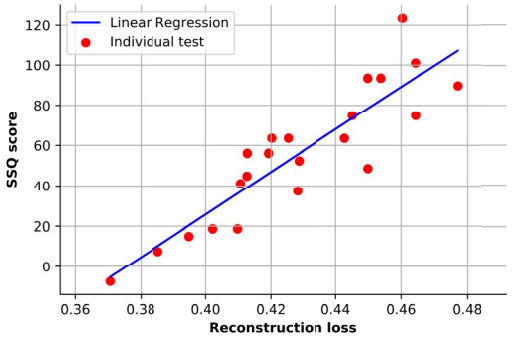

Figure 8: Correlation between the SSQ score and the loss computed from the model, regression line: $y = 1062.94x - 399.24$.

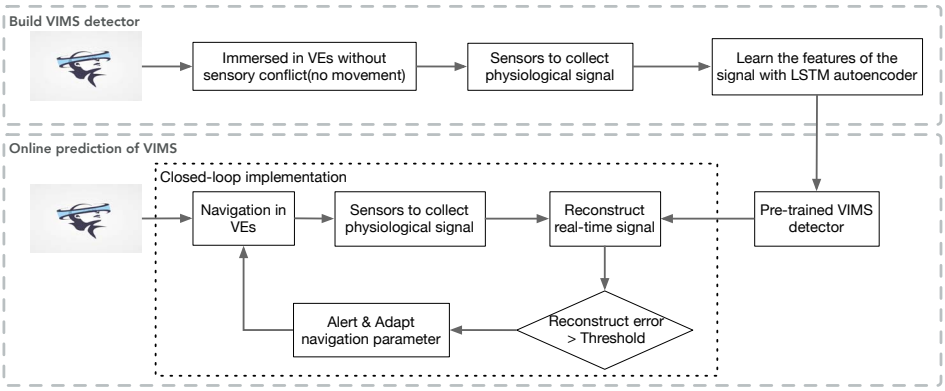

Figure 9: Schematic process of the proposed closed-loop system for navigation in VEs including a VIMS detector: the first step is to collect signals to train the model network and the second step allows to deploy the pre-trained detector in a real application.

7 Conclusion and Future Work

We propose a new metric where we take into account dynamic information of the postural sway signal and we extract its features using a long short term memory (LSTM) encoder; then, the level of VR sickness is measured according to the reconstruction error arising from the LSTM network. This method includes two steps in terms of implementation: firstly to train the network with signals from a normal state of the VR user and then to deploy the pre-trained model on a VR application. Since the model is only forced to learn the features when the user does not feel any VR sickness, it is possible for the system to detect and alert abnormal signals which is obtained when the user is in another physiological state and feels VR sickness. As we train independently the model for each user, the model is customized. The reliability of the network prediction was validated with the SSQ score and achieved high enough Pearson correlation coefficients, which means that the proposed method has the potential to be used in personalized VR navigation to detect cybersickness with real-time performance.

In this paper we only proposed the general method but we did not implement the whole system as depicted in Sect. 6. In the future, we plan to verify this method with other signals such as EEG, pupil dilation and saccades in time-series which are supposed to be more convenient for real time implementation and applications. In addition, it would be interesting to couple different physiological signals to see whether the prediction accuracy can be improved.

Acknowledgments

This work was supported by the China Scholarship Council under grant No. 201708390014.

References

[1] A. Alshaer, H. Regenbrecht, and D. O'Hare. Immersion factors affecting perception and behaviour in a virtual reality power wheelchair simulator. *Applied Ergonomics*, 58:1–12, 2017. doi: 10.1016/j.apergo.2016.05.003

[2] T. Aslandere, D. Dreyer, and F. Pankratz. Virtual hand-button interaction in a generic virtual reality flight simulator. In *2015 IEEE Aerospace Conference*, pp. 1–8, 2015. doi: 10.1109/AERO.2015.7118876

[3] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181

[4] J. Bolton, M. Lambert, D. Lirette, and B. Unsworth. Paperdude: A virtual reality cycling exergame. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pp. 475–478. ACM, New York, NY, USA, 2014. doi: 10.1145/2559206.2574827

[5] S. B. Brundage, J. M. Brinton, and A. B. Hancock. Utility of virtual reality environments to examine physiological reactivity and subjective distress in adults who stutter. *Journal of Fluency Disorders*, 50:85–95, 2016. doi: 10.1016/j.jfludis.2016.10.001

[6] J. R. Chardonnet, M. A. Mirzaei, and F. Mérienne. Features of the Postural Sway Signal as Indicators to Estimate and Predict Visually Induced Motion Sickness in Virtual Reality. *International Journal of Human-Computer Interaction*, 33(10):771–785, 2017. doi: 10.1080/10447318.2017.1286767

[7] W. Chen, A. Plancoulaine, N. Férey, D. Touraine, J. Nelson, and P. Bourdot. 6dof navigation in virtual worlds: Comparison of joystick-based and head-controlled paradigms. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, VRST '13, pp. 111–114. ACM, New York, NY, USA, 2013. doi: 10.1145/2503713.2503754

[8] S. Davis, K. Nesbitt, and E. Nalivaiko. A systematic review of cybersickness. In *Proceedings of the 2014 Conference on Interactive Entertainment*, IE2014, pp. 8:1–8:9. ACM, New York, NY, USA, 2014. doi: 10.1145/2677758.2677780

[9] L. Deng and D. Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3-4):197–387, June 2014. doi: 10.1561/2000000039

[10] M. S. Dennison, A. Z. Wisti, and M. D'Zmura. Use of physiological signals to predict cybersickness. *Displays*, 44:42–52, sep 2016. doi: 10.1016/j.displa.2016.07.002

[11] L. Dziuda, M. P. Biernacki, P. M. Baran, and O. E. Truszczyński. The effects of simulated fog and motion on simulator sickness in a driving simulator and the duration of after-effects. *Applied Ergonomics*, 45(3):406–412, 2014. doi: 10.1016/j.apergo.2013.05.003

[12] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya. Deep learning for healthcare applications based on physiological signals: a review. *Computer Methods and Programs in Biomedicine*, 161:1–13, 2018. doi: 10.1016/j.cmpb.2018.04.005

[13] L. Frank, R. S. Kennedy, R. S. Kellogg, and M. E. McCauley. Simulator sickness: A reaction to a transformed perceptual world. 1. scope of the problem. Technical report, ESSEX CORP ORLANDO FL, 1983.

[14] J. F. Golding. Predicting individual differences in motion sickness susceptibility by questionnaire. *Personality and Individual Differences*, 41(2):237–248, 2006. doi: 10.1016/j.paid.2006.01.012

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016.

[16] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 733–742, June 2016. doi: 10.1109/CVPR.2016.86

[17] J. L. Higuera-Trujillo, J. L.-T. Maldonado, and C. L. Millán. Psychological and physiological human responses to simulated and real environments: A comparison between photographs, 360° panoramas,

and virtual reality. *Applied Ergonomics*, 65:398–409, 2017. doi: 10.1016/j.apergo.2017.05.006

[18] J. Jankowski and M. Hachet. Advances in interaction with 3d environments. *Comput. Graph. Forum*, 34(1):152–190, Feb. 2015. doi: 10.1111/cgf.12466

[19] R. S. Kennedy, J. Drexler, and R. C. Kennedy. Research in visually induced motion sickness. *Applied Ergonomics*, 41(4):494–503, 2010. Special Section - The First International Symposium on Visually Induced Motion Sickness, Fatigue, and Photosensitive Epileptic Seizures (VIMS2007). doi: 10.1016/j.apergo.2009.11.006

[20] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, jul 1993. doi: 10.1207/s15327108ijap0303_3

[21] S. Khan and T. Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018. doi: 10.1016/j.ymssp.2017.11.024

[22] H. K. Kim, J. Park, Y. Choi, and M. Choe. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. *Applied Ergonomics*, 69:66–73, may 2018. doi: 10.1016/j.apergo.2017.12.016

[23] D. L. Harm. Physiology of motion sickness. In G. H. Crampton, ed., *Motion and Space Sickness*, pp. 153–178. CRC Press, Florida, 1990.

[24] S. M. Lavalle, A. Yershova, M. Katsev, and M. Antonov. Head tracking for the oculus rift. In *In IEEE International Conference on Robotics and Automation (ICRA). IEEE*, pp. 187–194, 2014.

[25] C. T. Lin, S. F. Tsai, and L. W. Ko. EEG-based learning system for online motion sickness level estimation in a dynamic vehicle environment. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10):1689–1700, 2013. doi: 10.1109/TNNLS.2013.2275003

[26] Y. MATSUURA and H. TAKADA. Evaluation of motion sickness induced by 3d video clips. *Nippon Eiseigaku Zasshi (Japanese Journal of Hygiene)*, 71(1):2–11, 2016. doi: 10.1265/jjh.71.2

[27] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks, Jr. Effect of latency on presence in stressful virtual environments. In *Proceedings of the IEEE Virtual Reality 2003*, VR '03, p. 141. IEEE Computer Society, Washington, DC, USA, 2003.

[28] N. Padmanaban, T. Ruban, V. Sitzmann, A. M. Norcia, and G. Wetzstein. Towards a Machine-learning Approach for Sickness Prediction in 360^o Stereoscopic Videos. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1594–1603, 2018. doi: 10.1109/TVCG.2018.2793560

[29] J. Plouzeau, J.-R. Chardonnet, and F. Merienne. Using cybersickness indicators to adapt navigation in virtual reality: a pre-study. In *IEEE Virtual Reality (VR)*, pp. 661–662, 2018. doi: 10.1109/VR.2018.8446192

[30] J. T. Reason. Motion sickness adaptation: a neural mismatch model. *Journal of the Royal Society of Medicine*, 71(11):819–829, 1978.

[31] G. E. Riccio and T. A. Stoffregen. An ecological theory of motion sickness and postural instability. *Ecological psychology*, 3(3):195–240, 1991.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[33] S. Wibirama and K. Hamamoto. Investigation of visually induced motion sickness in dynamic 3D contents based on subjective judgment, heart rate variability, and depth gaze behavior. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4803–4806. IEEE, aug 2014. doi: 10.1109/EMBC.2014.6944698

[34] S. D. Young, B. D. Adelstein, and S. R. Ellis. Demand characteristics in assessing motion sickness in a virtual environment: Or does taking a motion sickness questionnaire make you sick? *IEEE Transactions on Visualization and Computer Graphics*, 13(3):422–428, May 2007. doi: 10.1109/TVCG.2007.1041