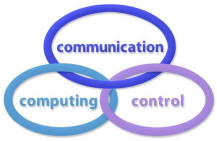
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 15, Issue: 2, Month: April, Year: 2020
Article Number: 3778, doi.org/10.15837/ijccc.2020.2.3778



CCC Publications

Prediction of Pending Data Using Interpolation and Extrapolation Techniques for Virtual Rowing

C. Canbulut, A. Paulauskas, T. Blazauskas

Cenker Canbulut
Department of Software Engineering
Kaunas University of Technology, Lithuania
Studentu g. 50, LT-51368, Kaunas, Lithuania
cenker.canbulut@ktu.lt

Andrius Paulauskas
Department of Software Engineering
Kaunas University of Technology, Lithuania
Studentu g. 50, LT-51368, Kaunas, Lithuania
andrius.paulauskas@ktu.lt

Tomas Blazauskas
Department of Software Engineering
Kaunas University of Technology, Lithuania
Studentu g. 50, LT-51368, Kaunas, Lithuania
tomas.blazauskas@ktu.lt

Abstract

The use of peripheral devices could be a good alternative to interact with Virtual Environment in addition to native controllers. Using combination of Concept-II peripheral device with VR mobile will cause latency issues due to different data transmission rates between those components. This latency issue leads to two major problems, such as micro-stutter in Virtual Environment (VE) and distance inaccuracy in time between the Concept-II peripheral device and VR mobile. In this paper, the authors present three algorithms based on interpolation and extrapolation methods, which aim to provide immersive VR experience by ensuring stutter-free and accurate rowing sessions for the user. This is relevance when considering the adoption of peripheral devices (i.e. the Concept-II peripheral device) as an alternative to interact with Virtual Environment in addition to native controllers. Predicting virtual rowing shell's position using interpolation by position method gives accurate time results but introduces high amount of micro-stutter. Using extrapolation method by taking speed parameter to predict rowing shell's position has very high time error but gives pleasant, stutter-free virtual rowing experience. Finally, adding this speed parameter a correction constant value for predicting virtual rowing shell's position provides stutter-free and very accurate in time rowing session for the user.

Keywords: virtual reality, data prediction, peripheral device, interpolation, extrapolation, latency.

1 Introduction

Virtual Reality uses computers to create 3D environments in which one can navigate and interact [6]. Today, virtual reality (VR) systems utilize more and more input and output media in order to make virtual environments more realistic. However, a user in a real-world environment experiences a higher set of feedback from real-world inputs, which relate directly to auditory, visual, and force feedback. As such, in a virtual environment, a dissociation is introduced between the user's inputs and feedback from the virtual environment [12]. This dissociation relates to the discomfort the user experiences with real-world interaction [12]. Common input devices such as mouse and keyboard, gamepads and touchscreens can be used with current VR systems to some extent, but are not considered intuitive or optimal solutions [20]. Interaction using these input devices occurs tactilely where the user performs required action on the device and sees the feedback through the virtual environment. In this approach, the user should be aware of the mouse position and synchronize him/herself depending on the head motion. This kind of self-synchronization forces users to perform inaccurate interactions and it often breaks the virtual immersion from the user perspective. This causes the need for a new type of VR development system architecture [1]. Because of these reasons, a lot of research and development is done to create or adapt input devices that collaborate for a variety of VR systems and their applications [20]. The motivation of this research is to discover new possibilities of using peripheral devices for controlling VR. For this purpose, the authors aim to adapt the Concept-II rowing machine into Virtual Reality to give an alternative way of controlling Virtual Environment. To perform successful rowing, a rower needs to obtain both physical and technical abilities [21]. By developing a VR solution dedicated to rowing, authors aim to create an alternative tool for novices and professionals to give an immersive experience for indoor training using VR mobile with Concept-II rowing machine.

Latency is defined as the amount of time a message to traverse a system [7]. It is also frequently cited shortcoming of VR applications[7][18][10]. Acceptable amounts for perceivable latency according to VR is under 20 milliseconds [7]. In our research case, the data generated with the Concept-II rowing machine has a transmission frequency of 10Hz which approximately corresponds to data transfer to VR application in every 100 millisecond. However, to achieve high-quality and smooth VR experience, the virtual reality application must operate at 60hz rate on mobile devices. Frequency incompatibility between these two devices leads to two significant problems to be solved.

- Time difference occurs between the completed rowing session (input time) and the time for the same session within the virtual environment due to pending distance data transferred from Concept-II to VR mobile.
- The micro-stutter problem appears due to latency, which causes the pending of data from Performance Monitor Bluetooth device to the smartphone.

Micro stutter is a term used in computing to describe the quality defect between frames rendered by GPU. This defect may happen because of different reasons and result in video stutters giving unpleasant gameplay experience in case of a video game. To solve these problems, the authors suggest using the proposed prediction methods. The methods are investigated using experimental studies and provided in this article.

2 Related works

The popularity of Virtual Reality (VR) had dramatically increased among consumers when Palmer Luckey introduced the prototype of Oculus Rift in 2010. Oculus Rift proved that it could depict VR of a Horizontal 90-degree field of view (FOV) [3]. Successes of the prototype over the next few years took the attention of other competitors on the market, such as Samsung, HTC, and Google. It didn't take very long for competitors to come up with their VR solutions on the market after the prototype of Oculus Rift.

Research "Reducing latency when using virtual reality for teaching sport" in 2008 has given suggestions on how heuristic prediction algorithms could be used to develop more effective and general systems for VR applications [10]. The problem presented in that article is closely related to the latency

or delay issue which is investigated in this research.One of the suggested prediction methods in the article was indicating that interpolation and extrapolation algorithms could be used to predict and give a pleasing experience for the user. By conducting this research, we validate the idea of using the studied methods in article [10].

During the development stage of the system, the authors of this research have elevated the requirement that the result of the rower should present the closest result as if the rower would perform it in a real environment. Data transmission between Concept-II and Mobile VR can prevent the correct result from being obtained due to different technical limitations such as data rate. An alternative example of this, when data transmits from a graphic card to a monitor. In this case, the output device (graphic card) sends frames out of sync with displays (monitor, HMD, etc.) refresh rate [2]. This desynchronization called "screen tearing" means a visual artifact occurring when image data corresponding to two or more different frames on a screen in a display. Today different methods can be applied to eliminate screen tearing problems such as V-Sync, Free-Sync, Fast Sync, or G-Sync.

In the literature, the problem of integrating different frequency domains solved in various research fields such as computer engineering, electrical engineering using interpolation and extrapolation techniques. The use of these techniques often shows differences in the mean of implementation, depending on the problem itself. In signal processing, the use of interpolation and extrapolation techniques performed when predicting missing signals from the signal samples [15].

In many cases, neural network techniques are used to predict data to achieve autonomous behaviors of any vehicle, device, software or hardware components. In most of those cases, accuracy is one of the important aspect that determines the quality measurement of an applied technique. In June 2018, "Autopilot Design for Unmanned Surface Vehicle based on CNN and ACO" uses predictive control method to reach defined destination with autonomous sailing depending on the sea conditions. A NPMC controller based on convolutional neural network(CNN) and ant colony optimizer(ACO), predict the disturbances to steer marine surface vehicle to have more comfortable journey to the destination [22]. Prediction of pending data in this research is open for further experiment using neural network techniques which may result in more accurate and stutter-free virtual rowing experience.

One of the most common control units that researchers find interest in is haptic feedback adapted to Virtual Reality. Haptic feedback is the imitation of the touch sensation when touching virtual objects. With haptic feedback, the user perceives the object as something that takes up volume. The hardware part of these enhancements looks easy on the eyes, but functionality still being improved as more researches performed. A common challenge in terms of functionality is called "frequency rate incompatibility". Frequency rate incompatibility often occurs due to different update rates between Virtual Reality devices and adapted devices itself. Research done on the adaptation of haptic feedback within virtual reality tries to predict the force value based on previous haptic force calculation using interpolation and extrapolation techniques. With this method, researchers provide smooth and accurate haptic force in high update rate during a low-frequency physical simulation of complex and deformable models [9]. As haptic feedback is an essential aspect of VR. In 2017 Pan and Niemeyer [16] had research on predicting thrown ball position and velocity using Unscented Kalman Filter to promote the idea of users interacting with dynamic physical objects in VE. The system predicts the future trajectory of the ball as it undergoes projectile motion, and renders a virtual scene to display the ball's location as well as any assistive cues [16]. During the early development years of VR, Polhemus Isotrak was often used as an orientation and position tracking device in virtual reality environments [4]. The common issue with the Polhemus Isotrak was the adaptation of the device into VR. Because of unsynchronized data frequency between VR and Polhemus Isotrak, many types of research were done to eliminate the latency (time delay) and jittering issues for the sake of smooth VR experience[4].A research conducted in Cambridge uses this sensor, which transmits user position and velocity data at 30hz frequency. To give convincing and appealing VR experience, the authors aim to synchronize the low-frequency data to application runtime using a Kalman filter. Filtered data then set to estimate the actual position and velocity within the VR application to provide appealing visualization in VE [5].

Calculating position of a controlled virtual object or a real object such as industrial robot may have similarities on how their position is determined in virtual world or real world. One of the prediction

algorithm in this article relies on the positioning of the boat in virtual environment. As this article deals with the linear interpolation, the complexity of prediction of an object becomes much harder task when non-linear problems are involved. In such scenarios, solutions as given in article [14] from different fields could be studied and adapted in virtual environment.

3 Implementation of the system

The single-user smartphone system consists of two main elements - a smartphone with a virtual boot system and a "Performance Monitor" computer. The VR Rowing application communicates and reads data from the Performance Monitor(PM) using Bluetooth Low Energy connection. This configuration requires the 5th Performance Monitor computer model, as this model currently the only one that supports Bluetooth wireless communication via "Bluetooth Low Energy".

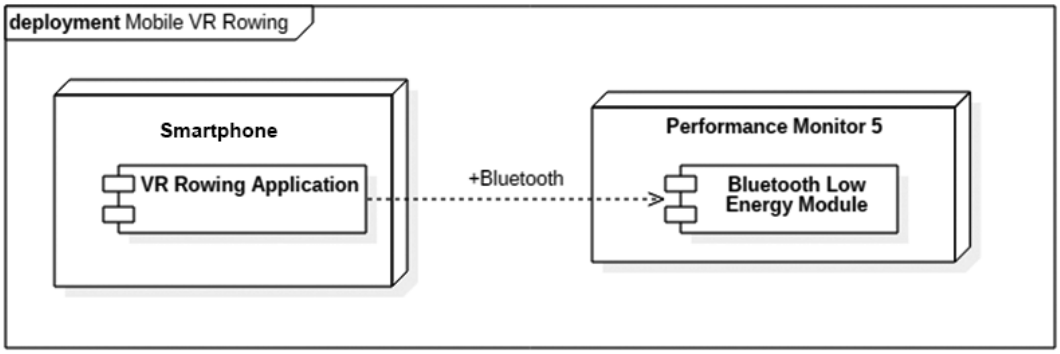


Figure 1: Chart of the routing system for the smartphone

During the implementation of the system, it was necessary to create a "Bluetooth" plug-in to use in the programming environment of Unity 3D Engine. The plugin lets Concept-II performance monitor to connect with the smartphone application to transfer data parameters.

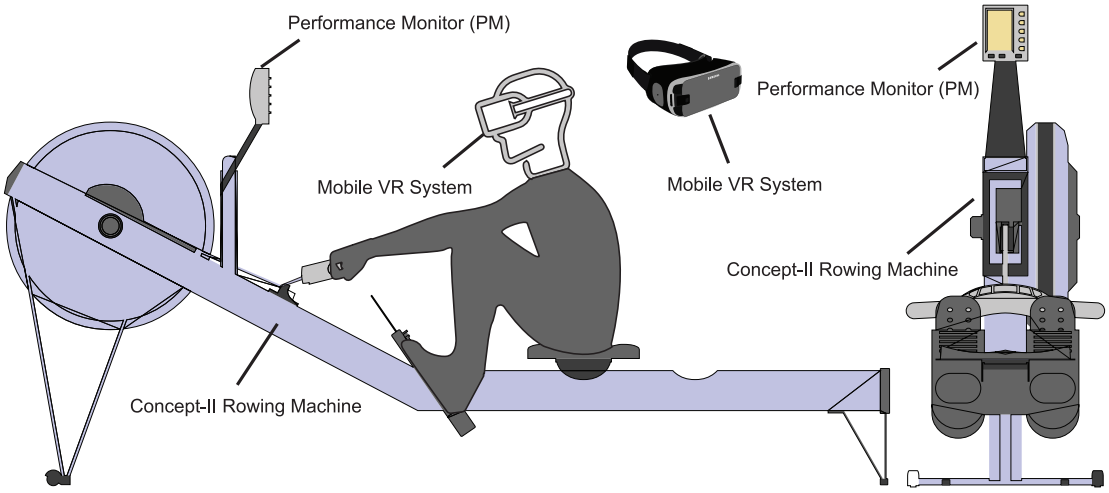


Figure 2: Virtual rowing simulator system components

The rowing workout data stored in the Performance Monitor (PM), which stands on the Concept-II peripheral device. This computer stores data in traveled meters, the average number of strokes per minute, average yield power, burnt calories, etc. All of these data can be transferred to other devices via the Bluetooth Low Energy (BLE) connection or through a USB interface using the CSAFE protocol. There are two essential parameters. One is the rowed distance, and the other is the state of the strokes. Rowed distance is used to move the virtual rowing shell, and the state of stroke used to animate the virtual rowing shell with rower's avatar within VE. Necessary data extracted by analyzing the characteristics of the "C2 rowing general". Characteristics stored in a 20-byte array.

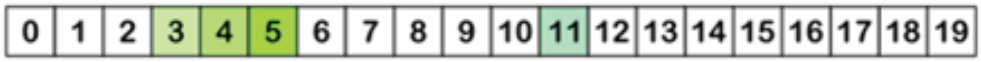


Figure 3: Rowing general status feature pack

The rowing distance is stored as a centimeter unit in the array in 3 bytes starting from 4th index. The stroke state stored in one byte at the 12th index. Stroke state has five statuses:

- Waiting until the rowing machine reaches the minimum speed (state value is 0).
- Waiting for the rowing machine to reach the speed of the simulator (state value is 1).
- Row driving state (state value is 2).
- Rower completes the row (status value is 3).
- Rower returns to start of row (status value is 4).

Performance Monitor updates the characteristics of the data. Each time the values are updated, they forwarded to the BLE connection on the smartphone. By default, PM updates values every 500 milliseconds. Sending frequency can be increased by adding specific value to the "C2 rowing general status and the additional status sample rate" characteristics. Meanings are:

- 0 - set the 1 second update period.
- 1 - set the 500ms update period (default).
- 2 - set the 250ms update period.
- 3 - set the 100ms update period.

As the VR rowing application needs the up-to-date data from the PM, it is convenient to change the default value of 1 and set it to 3, which corresponds to a 100 millisecond update period.

The system has three methods. One is prediction using linear interpolation by position. The other two methods are prediction using extrapolation by the speed where one of them uses an additional parameter to correct the speed. For this reason, it is relevant to show system flow within activity charts for two methods. One for the linear interpolation by position and another for extrapolation by speed with the correction.

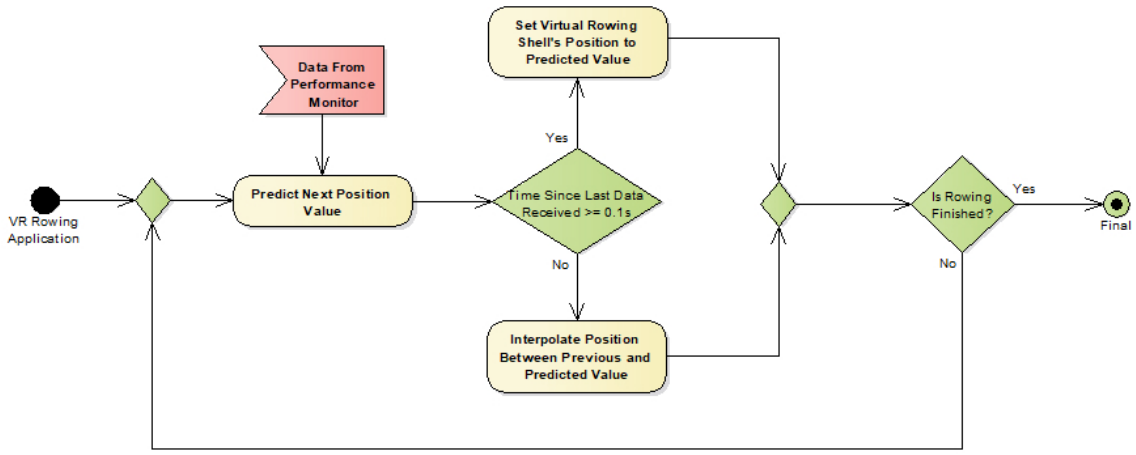


Figure 4: Activity Diagram of "Prediction using linear interpolation by position" method.

Data from Performance Monitor indicates the received data input from the Concept-II rowing machine. The system predicts the next position value by taking the difference in distance between the

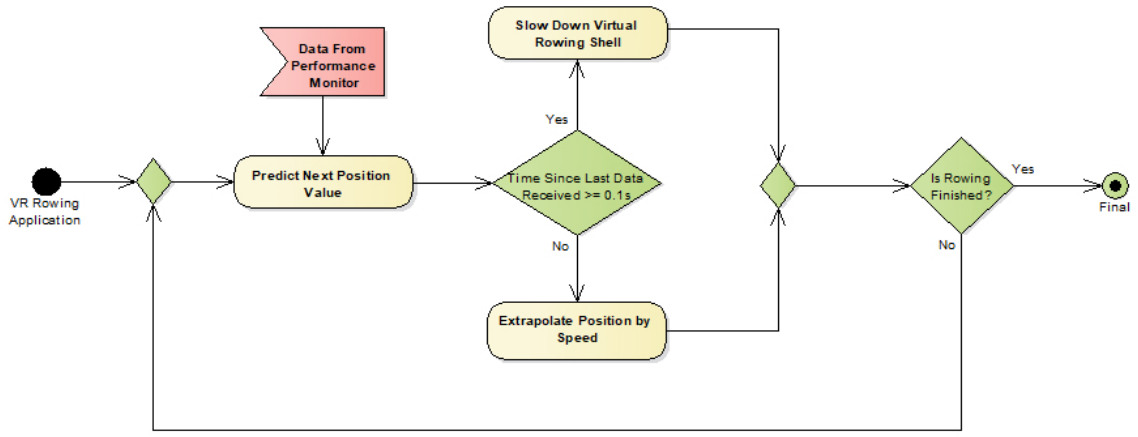


Figure 5: Activity Diagram of Prediction using extrapolation by correcting speed method.

current and previous frames. Then it checks if the last received data came at 0.1 seconds or longer. If the received data is under 0.1 seconds, it interpolates the position between previous and predicted values (see Fig. 4).

The system is using core principles the same way as the previous method, the difference comes on the decision of what happens if the received data came within 0.1 seconds or longer. Method slows down the boat if data received is above the constant and extrapolate it when it is opposite (see Fig. 5).

4 Methods to predict pending data in VR

Smartphones, in general, can refresh at a rate of 60 frames per second (FPS) [8]. PM can send data at 10Hz frequency to VR rowing application. This difference in frequency prevents VR rowing application to receive new data every frame and creates stutter effects during VR experience. The total number of frames rendered by the smartphone (VR mobile) defined as n, and the number of data packages sent from PM to the smartphone as m. l is the possible lag occurrence when sending data from PM to the smartphone.

$$n = [0, 1, 2, ...], \tag{1}$$

$$m = \left\lfloor \frac{n+l}{6} \right\rfloor, \tag{2}$$

from the given definition above, the data transition rate between the VR mobile application and Performance Monitor can be expressed in time (t) space, where t_i is time-stamp for VR mobile and t_j is the time-stamp of data coming from Performance Monitor.

$$t_i = [t_0, t_1, t_2..., t_n], i = [0, 1, 2, ..., n] \tag{3}$$

$$t_j = [t_1, t_2, t_3..., t_m], j = \left\lfloor \frac{i+l}{6} \right\rfloor. \tag{4}$$

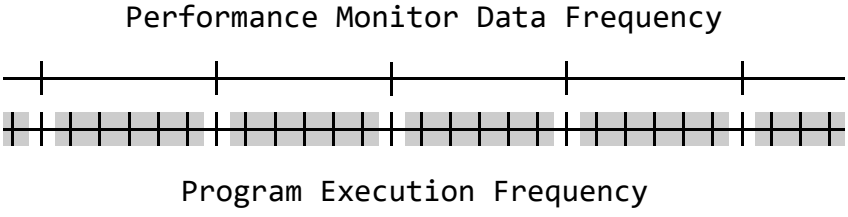


Figure 6: Performance Monitor Data Frequency versus Execution Frequency of Application.

The application renders in higher frequency than the data sent by PM (see Fig. 6). As the application waits for more data from PM (gray area in Fig. 6), the virtual rowing shell stays in standing position within VE until the new data comes. This discrete data transmission causes latency and creates stutter effects on VR mobile's VE. Proposed methods to predict pending data are aiming more accurate and stutter-free rowing sessions. To provide that, the authors suggest three prediction methods using linear interpolation and extrapolation techniques.

4.1 Prediction using linear interpolation by position

At the prediction phase of the rowing shell's position within the VE, it is necessary to get arbitrary value from PM to apply linear interpolation. The easiest way to predict the next frame's data is to assume that the data change in distance between current D_j and future D_{j+1} will be the same as the difference between existing D_j and past data D_{j-1}. The delta of the latter is calculated,

$$\Delta D_j = D_j - D_{j-1}, \tag{5}$$

and added to the current distance to get the predicted value,

$$D_{j+1} = D_j + \Delta D_j. \tag{6}$$

When calculating the predicted value, the rowing shell's position is interpolated from the previous position to the presumed position until the new data is received. The interpolation parameter is calculated during every application frame. The value is equal to the time that has passed since the last received data from PM divided by the data rate. If the time since last data from PM received is higher than the data rate, the interpolation parameter value is clamped at 1.

$$f(T) = \begin{cases} 1, & \text{if } t_i - t_j \geq 0.1 \\ \dfrac{t_i - t_j}{0.1}, & \text{otherwise} \end{cases} \tag{7}$$

Implementing this conditional function ensures that the virtual rowing shell's position will not travel further than the traveled distance in real-time on rowing machine.

$$BP_i = (1 - T) \times D_j + T \times D_{j+1}, \tag{8}$$

where, BP_i is the virtual rowing shell's position within VR mobile application.

The method requires an ideal data transfer of 0.1 sec. If data obtained is not equal at this ideal time, then micro stuttering may occur and can cause VR sickness [13] for the user during the VR experience, which may prevent the continuous and convincing illusion of virtual rowing shell movement. Because of this reason, the starting position of interpolation has to be the position that has interpolated up until the current time. Another issue may arise from incoming data delay. Because of this delay, interpolation may complete before the next data is received, which keeps the rowing shell in place for a few frames. Therefore, the authors searched and proposed other prediction methods.

4.2 Prediction using extrapolation by speed

Another approach of data prediction focuses on the virtual rowing shell's speed parameter rather than its position. In this approach, the method calculates the rowing shell's position difference between the current and the last frame and divides the result by the period of data transmission $TP = 0.1$ to obtain the rowing shell's current speed.

$$\Delta D_j = D_j - D_{j-1}, \tag{9}$$

$$v_j = \frac{\Delta D_j}{TP}. \tag{10}$$

To change virtual rowing shell's position BP in every frame within VR mobile application,

$$\Delta t = t_i - t_i - 1. \tag{11}$$

To predict the virtual rowing shell's position in the next frame, extrapolation function below is used,

$$BP_{i+1} = BP_i + v_j \times \Delta t. \tag{12}$$

This method provides continuous virtual rowing shell movement when the period of data received from the PM varies. But, if the rower's rowing tempo often changes during the rowing session, it increases the error by time between rowed distance in the PM and the virtual rowing shell's position within VE in the VR mobile application. Tempo change is a very frequent and natural occurrence in a rowing performance, that is why an additional solution had to be found to optimize the distance error between PM and the VR mobile application. The next proposed method decreases the error by far and gives less error margin compared to two methods described previously.

4.3 Prediction using extrapolation by speed with correction

Almost all prediction algorithms contain one or more parameters that are used for tuning to optimize performance, and a significant aspect in selecting an appropriate prediction algorithm depends on successfully adjusting an algorithm's parameter values [11]. In our case, to optimize the rowing shell's position in VE, we use a constant (C) value of 0.25, which modifies the speed parameter of the virtual rowing shell in each frame and attempts to correct it. This correction applies by increasing the speed with constant if the distance of the virtual rowing shell in the VR mobile application is lagging behind the data obtained from the PM, and reduce it if it exceeds.

$$f(BP_{i+1}) = \begin{cases} BP_i + v_j \times \Delta t(1-C), & \text{if } BP_i \geq D_j \\ \dfrac{t_i - t_j}{0.1}, & \text{otherwise} \end{cases} \tag{13}$$

In this approach, the position of the virtual rowing shell fluctuates around the input data and eliminates the error accumulation.

5 Experimental results

The purpose of the experiment is to ensure the effectiveness of proposed methods in terms of distance accuracy in time and stutter-free VR experience for the user. This experiment contains ten rowing sessions. Each session corresponds to 250 meters distance. Two factors determine the evaluation of the methods. One is the time difference after completing a session between VR mobile application and rowing machine. Another is micro-stutter measurement during the VR mobile experience. The effectiveness of the methods are calculated in each rowing session simultaneously at one rowing instance.

- Time difference after completing a session between two components caused because of the distance error occurrence within VR mobile application during a rowing session.
- Micro-stutter measurement is another aspect which has major impact on pleasing, smooth and continuous motion of the virtual rowing shell within VR mobile application. In this experiment, micro-stutter S is measured by checking if the distance of current frame D_i is same as the previous frame D_{i-1}.

$$f(S) = \begin{cases} S+1, & \text{if } D_i = D_{i-1} \\ S = S, & \text{otherwise.} \end{cases} \tag{14}$$

5.1 Results of finished rowing time difference between proposed methods and input data

The difference in time occurs because of distance error between VR mobile application and rowing machine. This distance error leads to time difference in the end of a rowing session.The session in VR mobile application might finish earlier or later than the actual rowing performed on rowing machine.

Table 1: Difference in time milliseconds (ms) after completing a session

Session No.	**Interp. by pos.**	**Extrap. by speed**	**Extrap. by speed with corr.**
1	+39.9	-6828	-62
2	+77.5	-8469	-18.4
3	+57.4	-7492	+0.2
4	+34	-4894	-5
5	+94.3	-8453	-63.7
6	+81.4	-6193	-14
7	+43.9	-6700	-33.5
8	+34	-7064	+16.6
9	+100	-4758	-43.3
10	+85.2	-7362	+32.8
AVG	+64.7ms	-6821 ms	28.95 ms

This difference in time depends on the method executed to predict input distance parameter. The negative sign in front of the number dedicates how much earlier in time the session finished compared to input time, where positive shows the opposite. The time difference is given in milliseconds.

Extrapolation by speed method accumulates distance error in time and this leads to very large error on time difference after a rowing session is completed. Interpolation by position has much less error in time, the maximum difference is on ninth session with 100 ms difference between actual finished time and the finished time within VR mobile application. Extrapolation by speed with correction method gives the least error after completing a rowing session with the maximum of -63.7 and minimum of 0.2 which is almost as same as the actual finished time coming from the rowing machine (see Table 1).

5.2 Results of micro stuttering during a rowing session

Micro stuttering measurement results rely on two variables. The number of stutters and average duration of each stutter during a session. As the total number of stutters increase, it is most likely to break VR immersion. If the amount of stutter is considerably high (100 - 200 stutters), it can lead to VR sickness [13]. Duration is another concern but less important than the stutter itself, it is just a measurement that shows average duration of a stutter. Depending on the stutter occurrence and amount, we can evaluate the impact of a method in terms of how it provides smooth VR experience for the user. Beside this, as it takes more time to finish a rowing session, it is most likely to have more stutters since the amount of frames will be more to render for GPU.

Table 2: Micros stutter count and average duration for each session with applied methods

			Methods		
		Input	In. by pos.	Ex. by spd.	Ex. by spd with corr.
S1	N of Stutters	478	263	0	0
	Avg. Duration	110.8 ms	37.9 ms		
S2	N of Stutters	524	289		
	Avg. Duration	111.2 ms	39.8 ms		
S3	N of Stutters	452	260		
	Avg. Duration	111.2 ms	37 ms		
S4	N of Stutters	426	227		
	Avg. Duration	111.1 ms	38.9 ms		
S5	N of Stutters	667	354		
	Avg. Duration	111.3 ms	38.2 ms		
S6	N of Stutters	447	238		
	Avg. Duration	111.3 ms	40 ms		
S7	N of Stutters	437	253		
	Avg. Duration	111.2 ms	37.8 ms		
S8	N of Stutters	535	273		
	Avg. Duration	111.2 ms	40 ms		
S9	N of Stutters	405	202		
	Avg. Duration	111.1 ms	40.8 ms		
S10	N of Stutters	448	247		
	Avg. Duration	111.1 ms	39.4 ms		

Input is the behavior of data transmitting from PM to VR mobile application with no methods applied. It gives an overview to understand how other proposed methods solve the micro-stuttering problem. From this overview, we can see that interpolation by position method decreases the problem by half, but still produces very high micro-stutter for any taken session. Extrapolation by speed and extrapolation by speed with correction eliminates the problem of micros stutter and gives smooth VR experience on any performed virtual rowing sessions on VR mobile (see Table 2).

6 Visualization of results

Every user's virtual rowing performance shows difference for given 250m distance long rowing track. As the system outputs the algorithm results for every frame, it is quite of a decision to represent the all sessions in one graph. Depending on the type of data that is obtained, the measurement data can take on a variety of representations [17]. Outputting all session results in one or separate graphs give very noisy result to distinguish one session from another. Because of this reason, two sessions were taken and indicated as"Slowest Rowing Session" (SRS) and "Fastest Rowing Session" (FRS) to see the characteristics between worst-case and best case scenario. The plotted data shows error distribution between the distance within VE on VR mobile and the actual input distance parameter from rowing machine. Each histogram plot distributes data using 32 bins. Graphs are outputted using Sturge's Rule [19]. The bins below zero shows how far behind the virtual rowing shell follows input distance parameter and bins above zero shows how far the virtual rowing shell traveled forward in distance compared to input distance parameter of rowing machine.

6.1 Error distribution of interpolation by position method compared to input distance data

Prediction linear interpolation by position shows left-skewed distribution (see Fig. 7). A large portion of data has distributed along 0.02 normalized values, and the probability factor has a higher occurrence between -30 to -5 intervals.

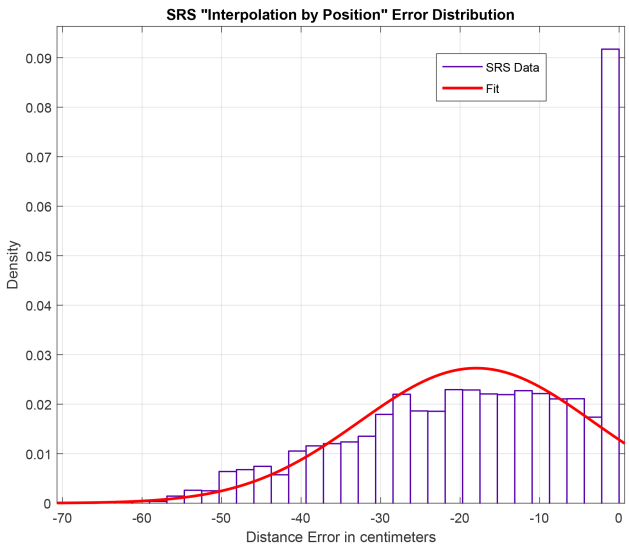


Figure 7: SRS - Error distribution of interpolation by position method.

Proposed method's distance is always following input distance parameter from behind. This means, the VR mobile application will finish the virtual rowing session after the actual session is completed. By looking at the given graph above, it is convenient to take two intervals and calculate the probability of distance error occurrence for the session. We take ranges of $-30 : -5$ and $0 : -5$ since the occurrence of data is higher at that ranges. The probability is 53.07% that the distance data within VR mobile will be on $-30 : -5$ centimeters behind the input distance parameter. And, the probability is 25.13%

that the distance data within VR mobile will be on $-5:0$ centimeters behind the input distance parameter.

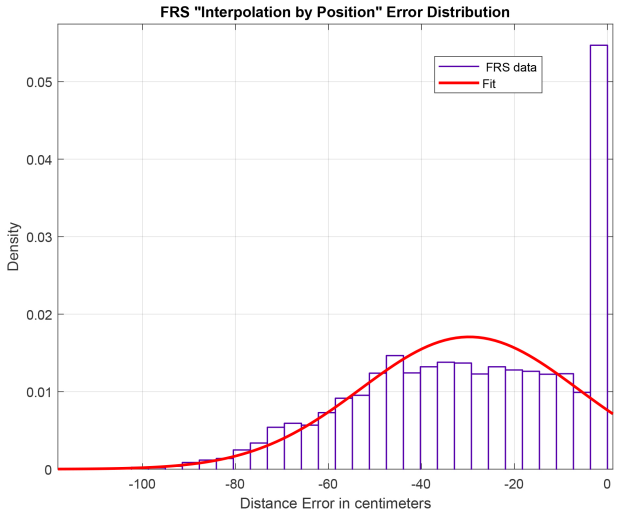


Figure 8: FRS - Error distribution of interpolation by position method.

FRS of interpolation by position method shows similar characteristics as in SRS (see Fig. 7 and Fig. 8). The skew is similar but the actual error occurrence is slightly higher than the SRS. From the graph (Fig. 8), it is convenient to take data range of $-50:-10$ and $-5:0$ since the occurrences of data is higher at that ranges. The probability is 58.18% that the distance data within VR mobile will be on $-50:-10$ centimeters behind the input distance parameter. And, the probability is 21.09% that the distance data within VR mobile will be on $-5:-0$ centimeters behind the input distance parameter.

6.2 Error distribution of Extrapolation by speed method compared to input distance data

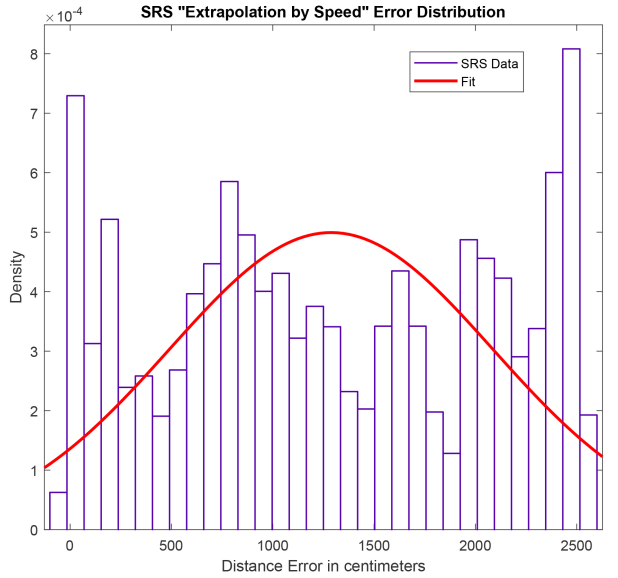


Figure 9: SRS - Error distribution of extrapolation by speed method.

During our experiment, this method gives volatile results in terms of how it predicts the input distance parameter on rowing machine. This method is unmodified version of the "extrapolation by speed with correction" method. In practice, these two methods showed significant differences on how the input distance data coming from rowing machine is predicted. Authors of this research believe that

plotting this model can give positive feedback for researchers on how a small but precise adjustment on an algorithm can give an effective outcome. As this model accumulates distance error exponentially by time, there is no actual benefit to output and make a comparison between SRS vs. FRS scenario. Therefore, only SRS scenario is outputted to show the performance overview of the method in one graph (see Fig. 9).

SRS of extrapolation by speed method shows that predicted distance values are distributed more or less evenly. It has no central tendency, and errors are substantial. The distance error accumulates exponentially as more distance data executed in time (see Fig. 9).

6.3 Error distribution of extrapolation by speed with correction method compared to input distance data

The extrapolation by speed with correction method shows that SRS data is symmetrically distributed and will have higher probability to row about 0:23 centimeters further than the input distance data (see Fig. 10).

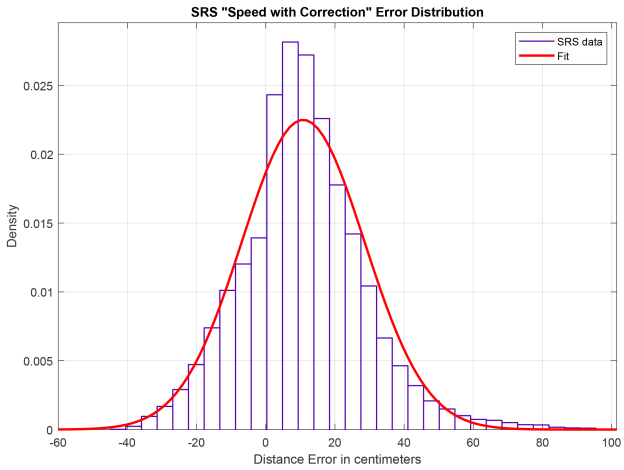


Figure 10: SRS - Error distribution of extrapolation by speed with correction method.

SRS of extrapolation by speed with correction method has normal distribution. This ensures more consistent and close proximity of input distance data prediction sent by the rowing machine. It is 70.19% probability that the data will likely be between $-20:20$ centimeters close to input distance data during a VR mobile application run-time. And, 38.69% probability that distance data in VR mobile application will be between $-10:10$ centimeters following the input distance data of rowing machine.

In the FRS graph (see Fig. 11), the distribution of error shows familiar characteristics as in the SRS counterpart (see Fig. 10). It has 51.2% of probability that data will likely be between $-20:20$ centimeters close to input data during a VR mobile application run-time. And, 25.42% probability that distance data in VR mobile application will be between $-10:10$ centimeters following the input distance data of rowing machine.

7 Conclusion

In this research, two problems were investigated as micro-stutter and distance error in time that breaks the VR immersion for the user. Problems were caused because of the different data transmission rates between the rowing machine and VR mobile. Three algorithms were proposed to solve this problem. The experimental research shows that:

- Interpolation by position method can provide accurate distance value during the virtual rowing session, always following the input distance parameter, but introduces a high amount of stutter, which can result in undesired, unpleasant VR experience. This breaks the VR immersion and

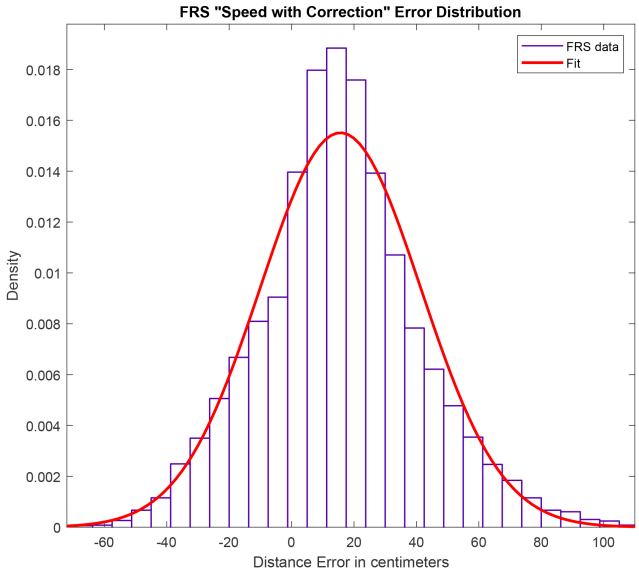


Figure 11: FRS - Error distribution of extrapolation by speed with correction method.

can cause VR sickness for the user. As the method always follows the input distance parameter, the actual time after completing a session will always be positive, meaning that the session will finish slightly later than the actual rowing done by the user in real-time.

- Extrapolation by speed method provides increasing distance error during the virtual rowing session. This causes the virtual rowing session to be finished much earlier than its actual finished time on the rowing machine. The error increases as the rowing session take more time to complete 250m. rowing distance. The presented algorithm provides a stutter-free rowing session, but doesn't result in a reasonable outcome.

- Correction added to extrapolation by speed method fixes the distance error and follows the input distance parameter much more accurately. The method gives closer proximity in the distance error range of -20 to 20cm., -10 to 10cm. with higher probability than other methods following input data distance parameter in time. This ensures a smaller time difference when a session is completed. The method provides a complete stutter-free VR experience for all sessions.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

References

[1] Black, R.; Landauer, J.; Rösch, A.; Simon, A. (1998). A highly flexible virtual reality system, *Future Generation Computer Systems*, 14(3-4), 167-178, 1998.

[2] Ceccotti, H.; Volosyak, I.; Gräser, I. (2010). Reliable visual stimuli on LCD screens for SSVEP based BCI, *18th European Signal Processing Conference*, IEEE, 919-923,2010.

[3] Desai, P.R.; Desai, P.N.; Ajmera, K.D., and Mehta, K. (2014). A Review Paper on Oculus Rift, *International Journal of Engineering Trends and Technology (IJETT)*, 13(4), 175-179, 2014.

[4] Emura, S.; Tachi, S. (1998). Multisensor Integrated Prediction for Virtual Reality, *Presence: Teleoperators and Virtual Environments*, 7(4), 410-422, 1988.

[5] Friedmann, M.; Starner, T.; Pentland, A. (1992). Synchronization in Virtual Realities, *Presence: Teleoperators and Virtual Environments*, 1(1), 139-144, 1992.

[6] Gutierrez, M.; Vexo, F.; Thalmann, D. (2008). *Stepping into virtual reality*, Springer, 2008.

[7] Gutmann, G.; Konagaya, A. (2019). Predictive Simulation: Using Regression and Artificial Neural Networks to Negate Latency in Networked Interactive Virtual Reality *Computer Science ArXiv 2019*, Tokyo, 2019.

[8] He, S.; Liu, Y.; Zhou, H. (2015). Optimizing Smartphone Power Consumption through Dynamic Resolution Scaling, *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*, 27-39, 2015.

[9] Hou, X.; Sourina, O. (2013). A prediction method using interpolation for smooth six-DOF haptic rendering in multirate simulation, *2013 International Conference on Cyberworlds, CW 2013*, IEEE, 294-301, 2013.

[10] Iskandar, Y.H.; Gilbert, L.; Wills, G.B. (2008). Reducing latency when using Virtual Reality for teaching in sport, *2008 International Symposium on Information Technology*, IEEE, 3, 1-5, 2008.

[11] Jung, J.Y; Adelstein, B.D; Ellis, S.R (2000). Discriminability of Prediction Artifacts in a Time-Delayed Virtual Environment, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 44(5), 499-502, 2000.

[12] Lambeta, M.; Dridger, M.; White, P.; Janssen J.; and Byagowi, A. (2016). Haptic wheelchair, *SIGGRAPH 2016 - AC SIGGRAPH 2016 Posters*.

[13] Lee, J.; Kim, M.; Kim, J. (2017). A Study on Immersion and VR Sickness in Walking Interaction for Immersive Virtual Reality Applications, *Symmetry*, 9(5), 78, 2017.

[14] Matica, L. M.; Oros, H. (2017). Speed Computation for Industrial Robot Motion by Accurate Positioning, *International Journal of Computers Communications & Control*, 12(1), 76-89, 2017.

[15] Michaeli, T.; Pohl, V.; Member, S.; Eldar, Y.C. (2011). U-Invariant Sampling : Extrapolation and Causal Interpolation From Generalized Samples, *IEEE Transactions on Signal Processing*, 59(5), 2085-2100, 2011.

[16] Pan, M.K.X.J.; Niemeyer, G. (2017). Catching a real ball in virtual reality, *IEEE Virtual Reality (VR)*, 4, 269-270, 2017.

[17] Papson, S.; Oagaro, J.; Polikar, R.; Chen, J.C; Schmalzel, J.L; Mandayam, S. (2004). A Virtual Reality Environment for Multi-Sensor Data Integration, *Proceedings of the ISA/IEEE Sensors for Industry Conference*, IEEE, 116-122, 2004.

[18] Russel, M.T.(2018). Virtual Reality System Concepts Illustrated Using OSVR *Published in the book VR Gems in 2018*, CRC Press, 2018.

[19] Sturges, H.A. (1926). The Choice of a Class Interval, *Journal of the American statistical association*, 21(153), 65-66, 1926.

[20] Velden, D.V. (2017). Touchpad in VR: Evaluating input devices in virtual reality, *Master's thesis*, Utrecht, 2017.

[21] Von Zitzewitz, J.; Wolf, P.; Novaković, V. (2009). Real-time rowing simulator with multi-modal feedback, *Sports Technology*, 1(6), 257-266, 2009.

[22] Zhao, D.; Yang, T.; Ou, W.; Zhou, H. (2018). Autopilot Design for Unmanned Surface Vehicle based on CNN and ACO, *International Journal of Computers Communications & Control*, 15(1), 429-439, 2018.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:
Canbulut, C.; A. Paulauskas, A.; Blazauskas, T. (2020). Prediction of Pending Data Using Interpolation and Extrapolation Techniques for Virtual Rowing, *International Journal of Computers Communications & Control*, 15(2), 3778, 2020.
https://doi.org/10.15837/ijccc.2020.2.3778