Virtual Reality (2020) 24:515–524
https://doi.org/10.1007/s10055-019-00416-7

ORIGINALPAPER

Comparing head gesture, hand gesture and gamepad interfaces for answering Yes/No questions in virtual environments

Jingbo Zhao[1,2] · Robert S. Allison[2]

Received: 13 November 2018 / Accepted: 27 November 2019 / Published online: 30 November 2019

Abstract

A potential application of gesture recognition algorithms is to use them as interfaces to interact with virtual environments. However, the performance and the user preference of such interfaces in the context of virtual reality (VR) have been rarely studied. In the present paper, we focused on a typical VR interaction scenario—answering Yes/No questions in VR systems to compare the performance and the user preference of three types of interfaces. These interfaces included a head gesture interface, a hand gesture interface and a conventional gamepad interface. We designed a memorization task, in which participants were asked to memorize several everyday objects presented in a virtual room and later respond to questions on whether they saw a specific object through the given interfaces when these objects were absent. The performance of the interfaces was evaluated in terms of the real-time accuracy and the response time. A user interface questionnaire was also used to reveal the user preference for these interfaces. The results showed that head gesture is a very promising interface, which can be easily added to existing VR systems for answering Yes/No questions and other binary responses in virtual environments.

Keywords Head gesture · Hand gesture · Virtual reality · Usability

1 Introduction

Recent improvements in sensor technologies have enabled human body movements to be accurately tracked in real time. With novel depth sensors such as the Kinect and the Leap Motion, a large volume of algorithms has been proposed and developed for body gesture (Lun and Zhao 2015) and hand gesture recognition (Cheng et al. 2016). The accurate and fast head tracking sensors in head-mounted displays (HMDs), such as the Oculus Rift DK2, also make real-time head gesture recognition possible (Zhao and Allison 2017) in addition to the systems that use cameras to track head movements (Morimoto et al. 1996; Terven et al. 2014). One possible application of gesture recognition is to integrate such algorithms into VR systems to interact with virtual worlds. A typical interaction scenario in VR systems is to answer Yes/No questions asked by virtual avatars or raised by VR systems. For instance, Abate et al. (2011) presented an augmented reality (AR)-based tour system that may require an interface for answering questions asked by virtual tour guides. Answering Yes/No questions in VR systems is usually done by buttons pressed on handheld devices or by using handheld devices to point to corresponding options in menus (as in the HTC Vive). However, potential problems for using handheld devices are that it may hinder or prevent users from using their hands to perform tasks, such as picking objects with their fingers or performing hands-free locomotion in virtual environments.

In the current study, we propose to use head gesture and hand gesture as alternatives to conventional gamepad interfaces or other interfaces that employ handheld devices to answer Yes/No questions in virtual environments. The head gesture interface and the hand gesture interface do not require the user to hold additional devices in their hands. This may give users much freedom in performing activities. Comparing to the hand gesture interface and the gamepad interface, the head gesture interface has a unique advantage

✉ Jingbo Zhao
jingbo@eecs.yorku.ca

Robert S. Allison
allison@eecs.yorku.ca

1 College of Information and Electrical Engineering, China Agricultural University, No. 17 Tsinghua East Road, Beijing 100083, China

2 Department of Electrical Engineering and Computer Science, York University, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

that it does not require extra devices for tracking head movements as head movements are directly tracked in VR systems that use head-mounted displays (HMDs). Similarly, CAVEs are usually equipped with tracking glasses or computer vision systems that monitor head movements. Thus, head movements data are readily available for head gesture recognition. To implement a head gesture interface, one would only need to integrate existing head gesture recognition algorithms into the VR system.

However, head gesture interfaces and hand gesture interfaces may have their own problems. For example, in head gesture interfaces, when responses are given by head movements tracked by HMDs or tracking glasses, the heaviness of the head worn devices may give people strain (Wille and Wischniewski 2015), typically neck pain, when performing rapid head movements such as nodding or shaking. Other issues include blurred images and increased simulator sickness when rapid head movements are made. In comparison, hand gesture interfaces may not be as physically demanding as head gesture interfaces since users' hands can be directly tracked by optical sensors without users holding any devices. However, these optical sensors usually have limited tracking range and the tracking performance will degrade when these sensors are interfered with other infrared (IR) devices in a VR system. On the other hand, gamepad interfaces are familiar devices to people. They may be preferred to other types of interfaces as people may have substantial experience using the gamepad interface.

The goal of the present study is to systematically evaluate and compare a head gesture interface, a hand gesture interface and a gamepad interface to answer Yes/No questions (or make other binary decisions) in virtual environments. The results of the study may help researchers or designers to select a suitable interface in their VR systems to answer such type of questions.

2 Related work

Previous work on the comparison of interaction techniques or interfaces to the conventional gamepad interface in VR systems primarily focused on virtual locomotion and navigation.

Nabiyouni et al. (2015) evaluated the Virtusphere technique, the real-walking interface and the gamepad interface. They showed that the Virtusphere as a moderate-fidelity technique was significantly outperformed by a high-fidelity real-walking interface and a well-designed low-fidelity gamepad interface as the Virtusphere was fatiguing and difficult to control due to its large inertia. Conversely, the real-walking interface was natural to people and the gamepad interface had a clear mapping between joystick movement and users' intended direction of travel, so it was easy to use.

Yan et al. (2016) proposed three types of finger gestures based on the Synaptics ForcePad for virtual walking. The proposed methods were compared to the gamepad interface in terms of their performance. The gamepad interface was found to have shorter task completion time but have problems in overshooting. It was also either too sensitive or not responsive compared to finger gesture interfaces.

Zielasko et al. (2016) evaluated five locomotion techniques. Among these techniques, the Adapted Walking in Place and the Accelerator Pedal involved lower limb movements. Leaning required upper body movements while seated. The Shake Your Head technique used only head movements tracked by an HMD. Similarly, these techniques were also compared with the traditional gamepad interface. They found that the Accelerator Pedal and the leaning technique performed better than other techniques in terms of user preference and task performance.

Cardoso (2016) compared a hand gesture interface based on the Leap Motion sensor, a gamepad interface and a gaze-based interface for locomotion in VR. Results showed that the hand gesture performed better than the gaze-based interface but worse than the gamepad interface.

Kitson et al. (2017) compared several seated leaning locomotion techniques to the joystick interface. They reported that participants in general preferred the leaning techniques as they are fun, engaging and more realistic, but the joystick interface was still easier to use and control.

More recently, Coomer et al. (2018) compared four locomotion methods, including the joystick interface, the Arm-Cycling, the Point-Tugging and teleporting. The Arm-Cycling is a locomotion technique that creates egocentric motion in VR based on the displacements of HTC Vive controllers held in users' hands when users perform cycling motion of their arms with the triggers on the HTC Vive being pressed down. The Point-Tugging is method that requires users to grab a virtual point in virtual environments by pressing the triggers on the HTC Vive controllers and then tug to move themselves in virtual environments, followed by releasing the triggers to complete the movement. They concluded that the Arm-Cycling was the best locomotion method among these four techniques as it gave better sense of spatial awareness and lower simulator sickness scores.

In addition, the work by Morency et al. (2007) is related to ours. They showed that head gesture was preferred to mice or keyboards by participants for responding confirmation questions prompted by dialog boxes in their experiments.

But to the authors' knowledge, there has been no research that evaluated and compared head gesture interfaces, hand gesture interfaces and gamepad interfaces to answer Yes/No questions in virtual environments.

3 Methods

The functionality of each interface and the associated ways to indicate Yes/No responses are shown in Table 1. We discuss the algorithm of each interface in detail in this section.

3.1 Head gesture interface

Zhao and Allison (2017) presented a real-time head gesture recognition algorithm on HMDs using cascaded hidden Markov models (HMMs) (Rabiner 1989). An HMM is governed by parameters including N the number of hidden states, M the number of observation symbols and the model parameter $\lambda = (A, B, \pi)$, where A is the matrix that represents the transition probability between states, B the matrix that represents the emission probability of a symbol observed from a specific state and π the initial state probabilities. The structure of the head gesture recognition framework (recapped in Fig. 1) consists of four components, which are the vector quantization model, the simple gesture layer, the complex gesture layer and the output selection module. The simple gesture layer has seven parallel left–right HMMs for recognizing simple gestures such as rotating left and rotating right, while the complex gesture layer has two parallel left–right HMMs for recognizing complex gestures: nodding and shaking. The Baum–Welch algorithm was used to train HMMs to obtain their respective model parameter λ, and the forward procedure was used to evaluate an observation sequence S, which consists of discrete symbols of quantized head angular velocities, using trained HMMs with their respective model parameter λ. During real-time operation, the vector quantization module reads head angular velocities from the HMD—the Oculus Rift DK2, and quantizes the continuous head angular velocities into discrete symbols. These discrete symbols are further buffered and fed into the simple gesture layer to determine whether a simple gesture exists in the buffered sequence. The outputs from the simple gesture layer are further buffered and fed into the complex gesture layer to determine whether a complex gesture has been made. Finally, the output selection module determines the final gesture using the outputs from the simple gesture layer and the complex gesture layer. We used the original implementation from the authors for the head gesture interface, and the implementation was trained using head gesture data collected from nineteen participants. The average accuracy for recognizing complex gestures was 98.5%. We only used three types of gesture outputs, including remaining still, shaking and nodding, for the head gesture interface. We ignored other head gesture outputs. For this interface, nodding represents Yes and shaking denotes No.

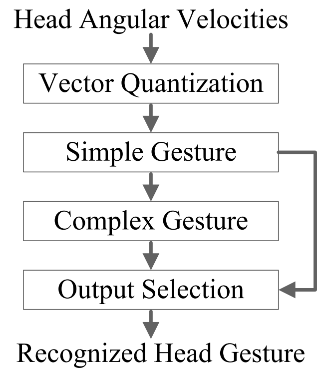


Fig. 1 Head gesture interface

3.2 Hand gesture interface

Marin et al. (2016) proposed a set of robust features for recognizing static hand gestures with the hand skeletons tracked by the Leap Motion sensor. The specific feature descriptor selected from the set for our implementation was:

$$P_i^x = \left(F_i - C\right) \cdot (n \times h)$$
$$P_i^y = \left(F_i - C\right) \cdot h$$
$$P_i^z = \left(F_i - C\right) \cdot n$$

where F_i is the position of the fingertip and i the index of a finger, C the position of the palm center, n the normal vector emanating from the palm and h the vector from the palm center to the direction of the fingers. These parameters are directly available from the tracked hand skeleton of the Leap Motion sensor. P_i^x, P_i^y and P_i^z are the extracted features. As pointed out by the authors, the set of equations normalize fingertip positions with respect to hand position and orientation. Fingertip angles, positions and elevations are embedded in the extracted features P_i^x, P_i^y and P_i^z. The extracted features can be used to train classifiers such as the support vector machine (SVM) (Chang and Lin 2011) to recognize static gestures.

Table 1 Definition of interface

	Yes	No	Standby
Head gesture	Nodding	Shaking	Head remaining still
Hand gesture	Waving an OK gesture	Waving an extended hand	Hand remaining still
Gamepad	Pressing button 5	Pressing button 6	Hand remaining still

Initially, we defined three types of hand gestures. The OK gesture represents Yes, and the extended hand gesture means No. The fist gesture was also defined as the standby gesture for resting (Fig. 2). A problem using only the static hand gesture recognition algorithm is that it is difficult to determine whether users intend to confirm their responses as static hand gestures are continuously recognized and a response can be determined before users finishing making their intended gestures. Thus, we included two HMMs to monitor the trajectory of the hand velocity to detect whether users are waving their hands or not. The outputs from the SVMs and the HMMs are fused by a set of rules to generate the final gesture: if the user waves a hand with an OK gesture, then the algorithm will confirm that the response from the user is Yes; similarly, if the user waves an extended hand, the response will be confirmed as No; otherwise, the algorithm considers that there are no meaningful responses given by users. Therefore, the types of hand gestures in the hand gesture interface were extended to six types, including: static OK gesture, static extended hand, static fist, waving OK gesture, waving extended hand and waving fist. The structure of the hand gesture recognition algorithm is illustrated in Fig. 3.

We collected hand gesture samples from twelve participants (age: 20–33, 7 males, 5 females) for the six types of gestures. Each participant was asked to perform four sessions of data collection. In each session, a participant was asked to perform the six types of hand gestures, respectively, and each type of hand gesture was recorded for 4 s. The features P_i^x, P_i^y and P_i^z extracted from the collected samples were used to train three SVMs using the one against one approach. To recognize a gesture during real-time operation, the voting strategy was used, meaning that the type of gesture that received the highest number of the votes given by SVMs is the winner. The average accuracy for recognizing static hand gestures is 99.6%. Two HMMs were trained using hand velocity data calculated from the hand centers. Specifically, one HMM was trained using hand velocities of static gestures, while the other was trained using hand velocities of waving hands. During operation, the HMM that produces the highest posterior probability when evaluating hand velocity data with their respective model parameter λ was chosen as the winner and its class label was given as the output. The average accuracy for recognizing whether a hand is waving or not is 98.1%. The theoretical average accuracy for recognizing dynamic hand gestures was 97.7% by multiplying the average accuracy for recognizing static hand gestures (99.6%) and the average accuracy for detecting moving hands (98.1%).

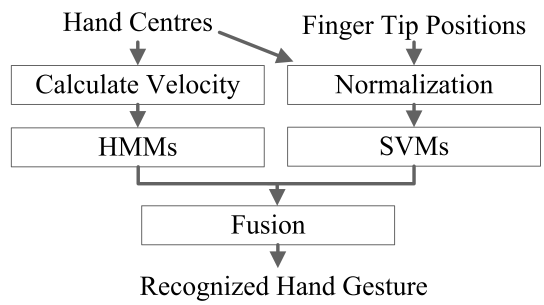


Fig. 3 Hand gesture interface

An advantage of the proposed hand gesture recognition framework is that it can be further extended to recognize combinations of different static hand gestures and different shapes of hand velocity trajectories. Thus, it has the potential to deal with more complex gesture recognition scenarios, but the average accuracy may decrease when more gestures are added.

3.3 Gamepad interface

The gamepad interface (Fig. 4) was implemented based on a Logitech Dual Action gamepad. Specifically, users pressed button 5 on the gamepad for Yes and pressed button 6 for No.

Fig. 2 Hand gestures (OK gesture, extended hand and standby)

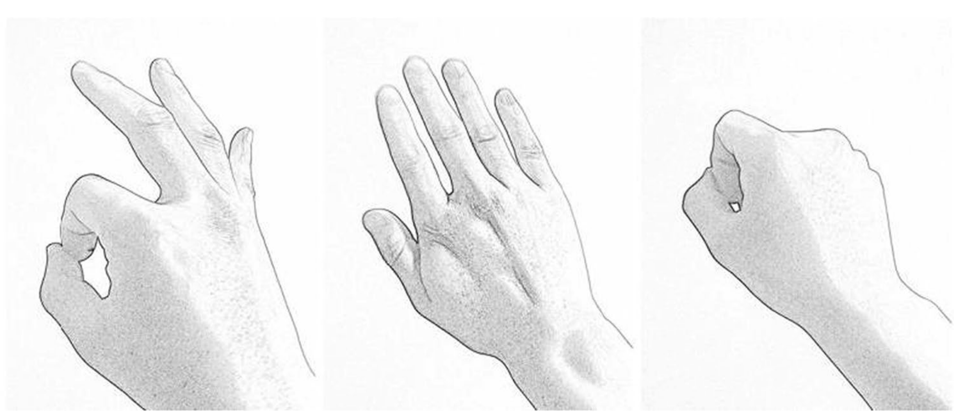


Fig. 4 Gamepad interface

4 Experiment

4.1 Introduction

The goal of the experiment was to evaluate and compare performance and user preference of the head gesture interface, the hand gesture interface and the gamepad interface for answering Yes/No questions in virtual environments. To achieve the goal, a memorization task was designed. The task asked participants to memorize the objects presented in a virtual room with a 30-s exposure period. Then, these objects were removed, and participants were asked whether they saw a specific object by answering Yes or No through a given interface. In Fig. 5a–c, we show the three stages of the experiment, including initialization stage, memorization stage and question stage. When a participant made a response, a confirmation (Yes/No) was prompted as shown in Fig. 5d.

4.2 VR hardware and software

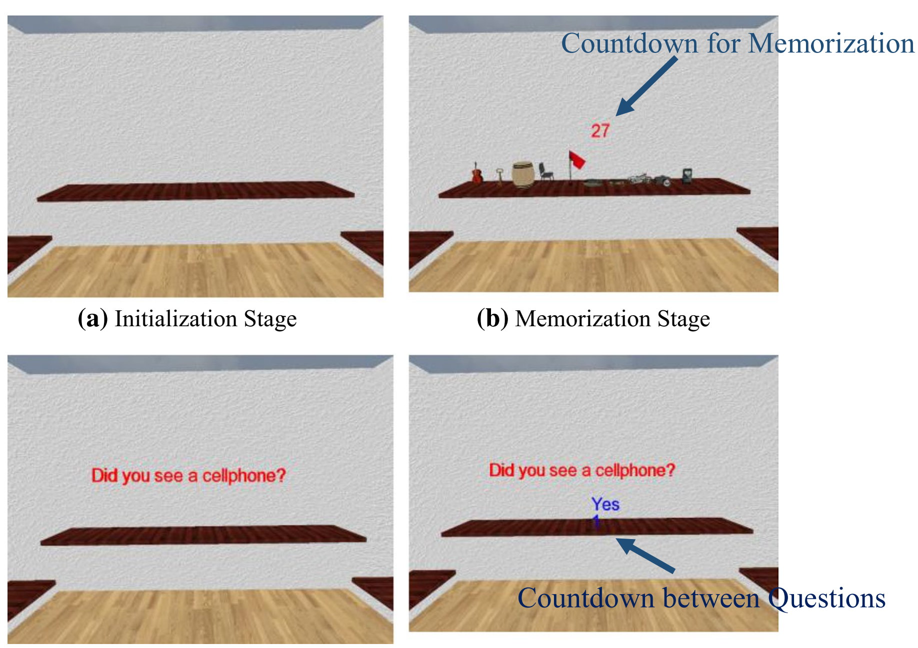
The host machine for the experiment was a desktop computer equipped with an Intel i7 2.8 GHz. CPU, 4 GB memory and an AMD Radeon HD 6850 graphics card. The operating system was Windows 7. Other experimental devices included an Oculus Rift DK 2, a Leap Motion sensor and a Logitech Dual Action gamepad connected to the host machine.

The experimental application and the algorithms for three interfaces were implemented in the Worldviz Vizard 5.0 using Python 2.7. The training of HMMs was done in MATLAB as it offered an HMM library convenient to use. Model parameters of HMMs were obtained after training. For real-time application, we only needed to implement the forward procedure of HMMs using Python in Vizard with the model parameters to evaluate movement sequences captured in real time. The support vector machine used in our study came from the LibSVM library (Chang and Lin 2011). Similarly, we also conducted training in MATLAB and then performed real-time application using the testing function of SVM with Python in Vizard.

4.3 Metrics

The metrics consisted of objective measures and subjective measures. The objective measures were: response time,

Fig. 5 Experimental stages



(a) Initialization Stage **(b)** Memorization Stage

(c) Question Stage **(d)** Response Made which is the time interval between when a question was prompted and when a response was made, and real-time accuracy, which is the percentage of the objects that were correctly classified as present or not during real-time operation. The objective measures were applied on the recorded experimental data to extract the corresponding parameters.

The subjective measures were: ease-to-learn, ease-to-use, natural-to-use, fun, tiredness, responsiveness and subjective accuracy. The subjective measures were evaluated using a user interface questionnaire modified from the one by Nabiyouni et al. (2015). The items given in the questionnaire are shown in Table 2. The seven-point Likert scale (from strongly disagree to strongly agree) was used to rate each factor.

4.4 Procedure

Informed consent was obtained from all twelve participants (age: 20–38, 7 males, 5 females) in accordance with a protocol approved by the Human Participants Review Subcommittee at York university. Twelve participants were divided into six groups. Each group covered a permutation of the gesture interfaces. Thus, six groups covered all six permutations of the three gesture interfaces. For each interface, four trials were conducted. The first trial was for training and was not considered for the data analysis, while the remaining three trials were the actual experiments. During experimental sessions, participants wore the Oculus Rift DK2 and sat 60 cm in front of the computer monitor, on which the tracking camera of the Oculus Rift DK2 was mounted. The Leap Motion sensor was attached onto a stand and was placed 30 cm in front of the participants. At the beginning of a trial, a participant was exposed to a virtual room, facing a bench placed in the front of the room. After the researcher pressed the start button, twenty objects were randomly selected from a list of thirty objects. The list consisted of everyday objects, including a camera, a cell phone and a chair, etc. The 3D models of the objects were obtained from www.turbosquid.com. Among the selected objects, ten were placed onto the bench with a random order and participants were given 30 s to memorize the presence of these objects. Another ten objects were not presented in the room and were only used for generating questions. After the 30-second memorization period, the presented objects were removed from the room. Participants were sequentially asked about the existence of the twenty objects with a random sequence. Questions had the form “Did you see a cell phone?” (Fig. 5c). Each time the participant made a response through the given interface; a 3-s waiting period was introduced before the next question was prompted. The object names, the timestamps when the questions were prompted and the timestamps when the responses were made, the existence of objects and the responses of the participants were recorded for data analysis. After participants completed four trials for a given interface, they were asked to complete the user interface questionnaire to evaluate the interface they used.

Table 2 The user interface questionnaire

1.	The interface is easy to learn
2.	The interface is easy to use
3.	The interface is natural and intuitive to use
4.	The interface helps make the task fun
5.	Using the interface is tiring
6.	The interface helps me respond quickly
7.	The interface helps me make accurate responses

4.5 Results

We performed the data analysis in MATLAB 2016a and R 3.4.2. One-way repeated-measure ANOVAs were applied on each factor of the objective measures and subjective measures to reveal whether there were significant effects between the types of interfaces. Post hoc pairwise comparisons were made using Tukey’s range tests.

We found a significant effect on response time ($F(2, 22) = 50.84,\ p < 0.001$). On average, the head gesture interface had the highest response time and the gamepad interface had the lowest, while the hand gesture was in the middle. A Tukey’s range test confirmed that the gamepad interface was significantly faster than the head gesture interface and the hand gesture interface in terms of response time and there was no significant difference between the head gesture interface and the hand gesture interface (Table 3). The results (Fig. 6, bars denote the mean value and error bars denote the standard error of the mean) were expected since the head gesture interface required nodding or shaking for at least one cycle. This typically took longer time than pressing a button on the gamepad or waving hands in front of the Leap Motion sensor. We also found a significant effect on real-time accuracy ($F(2, 22) = 16.70,\ p < 0.001$). A Tukey’s range test showed that the hand gesture interface was significantly less accurate than the gamepad interface and the head gesture interface (Table 3). The real-time accuracies of the three interfaces are shown in Fig. 7. (Bars denote the mean value, and error bars denote the standard of the mean.) The factor is primarily determined by the accuracy of memorization of the objects, the control of the interfaces and the recognition performance of the interfaces. Although, in theory, the gamepad interface should have the best recognition performance as Yes/No is recognized by two buttons, we found the head gesture had a slightly higher real-time accuracy than the gamepad interface with the assumption that the memorization of objects by participants across three interfaces was the same. This suggested that using the head

Table 3 Results of the Tukey's range tests on factors with significant effects (95% confidence level)

Factor	Interfaces	Difference	P value	Lower bound	Upper bound
Response time	Head versus gamepad	0.71	0.00	0.51	0.90
	Hand versus gamepad	0.65	0.00	0.46	0.85
	Hand versus head	− 0.05	0.77	− 0.25	0.14
Accuracy	Head versus gamepad	0.02	0.36	− 0.02	0.06
	Hand vs gamepad	− 0.07	0.00	− 0.11	− 0.03
	Hand versus head	− 0.09	0.00	− 0.13	− 0.05
Ease-to-use	Head versus gamepad	− 1.00	0.13	− 2.23	0.23
	Hand versus gamepad	− 1.83	0.00	− 3.07	− 0.60
	Hand versus head	− 0.83	0.23	− 2.07	0.40
Tiredness	Head versus gamepad	1.67	0.03	0.14	3.19
	Hand versus gamepad	1.33	0.09	− 0.19	2.86
	Hand versus head	− 0.33	0.85	− 1.86	1.19

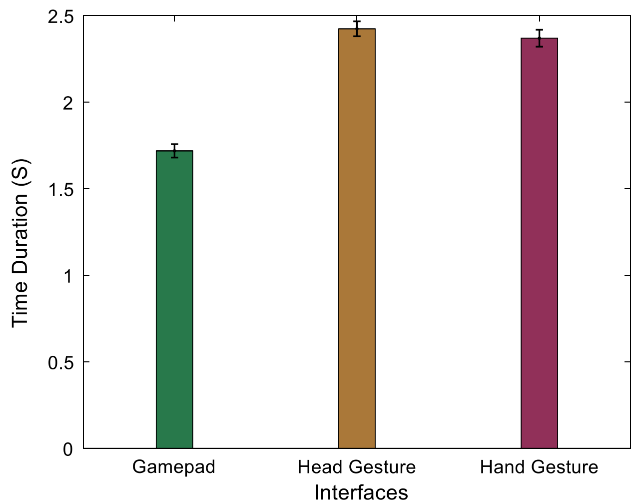


Fig. 6 Response time

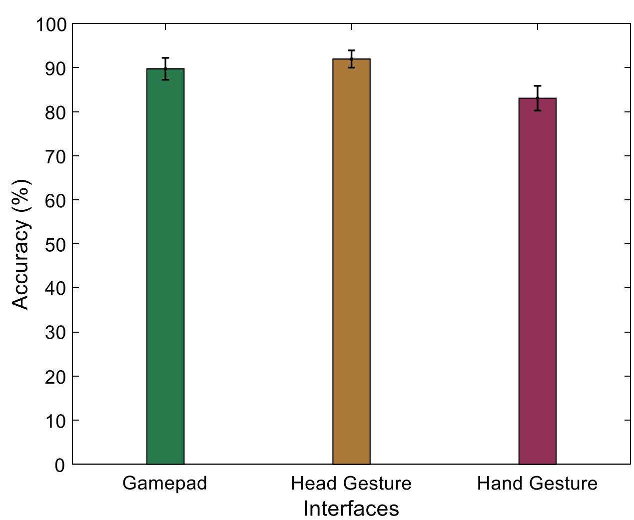


Fig. 7 Real-time accuracy

gesture interface was less error-prone than the gamepad interface. The hand gesture interface had the lowest real-time accuracy, which suggested that using hand gesture interface might introduce more errors into responses.

For subjective measures (Fig. 8, bars denote the mean value, and error bars denote the standard error of the mean), the gamepad interface was rated better than other interfaces in terms of ease-to-use, fun, tiredness, responsiveness and subjective accuracy, while the head gesture was rated slightly higher for ease-to-learn and natural-to-use. The hand gesture interface was not preferred for all factors except tiredness as the head gesture interface was considered as the most tiring interface. We found a significant effect on ease-to-use ($F(2, 22) = 7.00,\ p = 0.004$), and a Tukey's range test showed that the gamepad was significantly easier to use than the hand gesture interface (Table 3). A significant effect was also found on tiredness ($F(2, 22) = 4.22,\ p = 0.028$), and a Tukey's range test showed that the gamepad interface was significantly less tiring than the head gesture interface (Table 3). However, we did not find significant effects on other factors: ease-to-learn ($F(2, 22) = 2.27,\ p = 0.13$), natural-to-use ($F(2, 22) = 2.18,\ p = 0.14$), fun ($F(2, 22) = 0.65,\ p = 0.53$), responsiveness ($F(2, 22) = 2.89,\ p = 0.08$) and subjective accuracy ($F(2, 22) = 2.84,\ p = 0.08$).

One interesting finding was that responsiveness and subjective accuracy in the subjective measures did not agree with response time and real-time accuracy in the objective measures, respectively. For example, although subjectively participants indicated that the gamepad interface was more accurate than the head gesture interface, this was not the case when the accuracy was assessed objectively. Similarly, the head gesture interface took the longest time for making responses on average in the objective measure, but participants indicated that the hand gesture interface was less responsive than the head gesture.

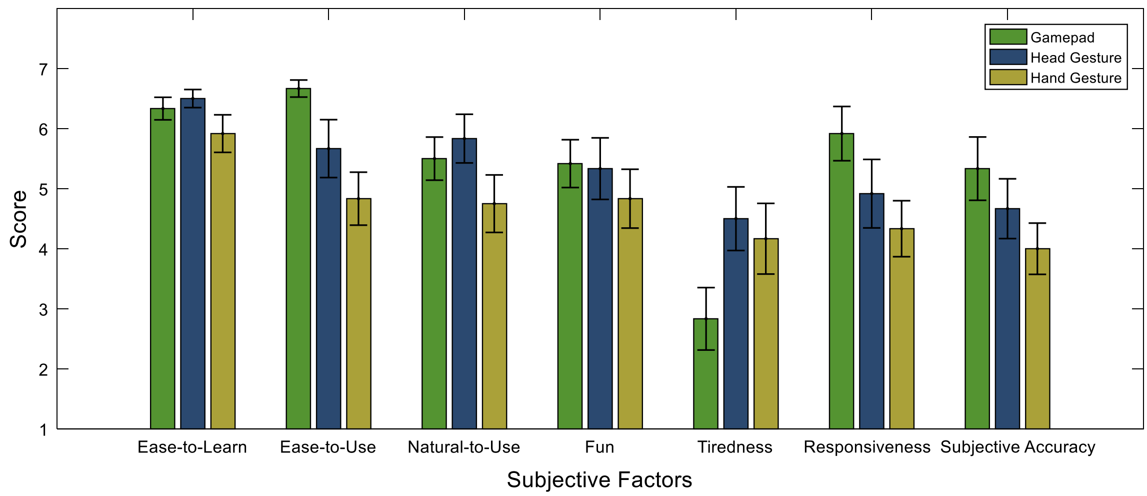


Fig. 8 Subjective measures

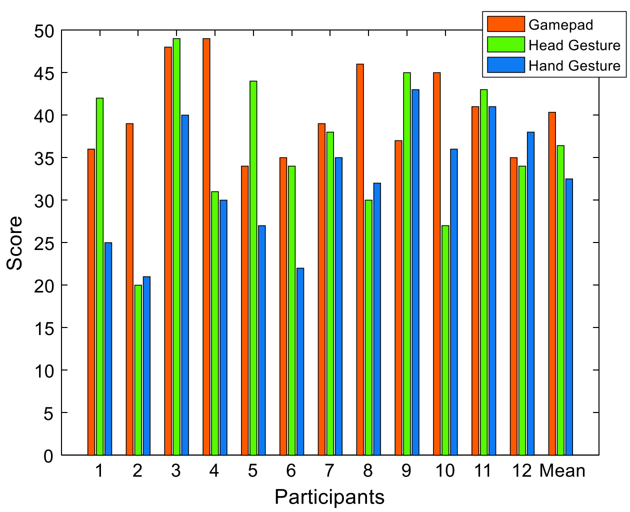


Fig. 9 Total score

The total scores of each interface rated by all participants are shown in Fig. 9. The score for the factor tiredness was inverted (strongly agree received one point and strongly disagree received seven points) to indicate how positive participants' attitudes were toward tiredness. We found that the gamepad interface was preferred by six participants, while the head gesture interface was preferred by five participants. Only one participant opted for the hand gesture interface. Thus, in general, the gamepad interface and the head gesture interface were equally liked, while the hand gesture was rejected. But in terms of the mean scores across participants (the 13th set of bars in Fig. 9), the gamepad interface still received the highest score.

5 Discussion

Gamepads or other handheld devices are traditional interfaces for people to play games and have a long history for interaction in VR systems. For example, joysticks have been used for flying in virtual environments as a method for locomotion (Robinett and Holloway 1992). Because of people's familiarity and previous experience with these devices, it is possible that even when new interfaces appear, they would still prefer these traditional devices as such devices may be more reliable. In addition, handheld devices are more familiar and would not take extra efforts for people to learn how to use them. In Fig. 9, six participants preferred the gamepad interface to other two interfaces. This showed that gamepads or handheld devices are still important devices for VR interactions.

Responding Yes/No through head nodding and shaking is a natural way for the interaction between people in the real world. In Fig. 8, the rating of natural-to-use was higher than other interfaces. Similarly, the interface was rated easier to use than other interfaces. As has been discussed, the primary problem with the interface is the heaviness of the HMD, which probably made people consider the head gesture interface the most tiring one. We expect that by using an HMD or tracking glasses with lower weight or using computer vision systems for tracking, the tiredness for using the interface would be lowered. But given tiredness as the primary limitation, the interface was still preferred by five participants.

The hand gesture interface was only preferred by one participant probably because the definition of Yes/No using a waving OK gesture and a waving extended hand was not natural or unfamiliar to participants. To make a response, the hand of a participant needed to make a two-step movement. First, they need to make an OK gesture or extend their hands. Then, they need to wave their hands to confirm their responses. It is obvious that more efforts are required when using the hand gesture interface than other two interfaces, which required only a one-step movement, such as pressing a button or shaking their heads. As shown in Fig. 8, the hand gesture was the most difficult to learn and most difficult to use. The factors fun, responsiveness and subjective accuracy were also lower than other two interfaces. It was only considered better than the head gesture interface in terms of tiredness. We expect that improved gestures for Yes and No might improve the usability of the hand gesture interface.

Finally, another option to implement the functionality to answer Yes/No questions in VR systems is to use speech recognition algorithms as interfaces to recognize people's voice. The performance and user preference of the speech recognition interface also can be studied and compared to the motion-based interfaces presented in this paper. In practice, we can also design a multi-modal interface that integrates the head gesture interface, the hand gesture interface, the gamepad interface and the speech recognition interface into a single system and let users choose their preferred interface during actual usage.

Limitations of the experimental design were that extra time was taken for participants to recall the objects they memorized when responding to Yes/No questions and the ability of the participants to memorize the given objects might also affect the results of the real-time accuracy. In addition, we had a sample of twelve participants who were young university-educated students. They might not be the representative of the general population or specific subpopulations such as children or the elderly. Future research should consider including more participants from diverse groups to determine whether the interface preferences found here generalize to these groups.

The research is important in the way that it provides VR researchers and designers with an idea of user preference and performance when users answering Yes/No questions or other binary questions using a head gesture interface, a hand gesture interface and a gamepad interface. Designers may consider several factors such as cost and the factors we studied in the paper when they do actual design and make decisions to include a specific interface or provide all three interfaces in their systems.

6 Conclusions

In this paper, we proposed to use the head gesture interface and the hand gesture interface to answer Yes/No questions in virtual environments. We evaluated their performance and user preference through a memorization task and compared them to the traditional gamepad interface. We showed that the head gesture interface was comparable to the gamepad interface. As adding the head gesture interface to a VR system usually does not require additional tracking devices, we suggest adding the head gesture interface to VR systems that require users to answer Yes/No questions. These techniques could also be readily adapted for other common binary responses (e.g., left versus right). We believe that interaction techniques using head gestures and hand gestures in VR systems are still underexplored. Thus, the utility of these interfaces in VR systems is worth further investigation.

References

Abate AF, Acampora G, Ricciardi S (2011) An interactive virtual guide for the AR based visit of archaeological sites. J Vis Lang Comput 22:415–425

Cardoso JCS (2016) Comparison of gesture, gamepad, and gaze-based locomotion for VR worlds. In: Proceedings of the 22nd ACM conference on virtual reality software and technology, pp 319–320

Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 27(2):1–27

Cheng H, Yang L, Liu Z (2016) Survey on 3D hand gesture recognition. IEEE Trans Circuits Syst Video Technol 26:1659–1673

Coomer N, Bullard S, Clinton W, Williams B (2018) Evaluating the effects of four VR locomotion methods: joystick, arm-cycling, point-tugging, and teleporting. In: Proceedings of the 15th ACM symposium on applied perception, pp 7:1–7:8

Kitson A, Hashemian AM, Stepanova ER, Kruijff E, Riecke BE (2017) Comparing leaning-based motion cueing interfaces for virtual reality locomotion. In: 2017 IEEE symposium on 3D user interfaces, pp 73–82

Lun R, Zhao W (2015) A survey of applications and human motion recognition with microsoft kinect. Int J Patt Recogn Artif Intell 29:1555008

Marin G, Dominio F, Zanuttigh P (2016) Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. Multimed Tools Appl 75:14991–15015

Morency L-P, Sidner C, Lee C, Darrell T (2007) Head gestures for perceptual interfaces: the role of context in improving recognition. Artif Intell 171:568–585

Morimoto C, Yacoob Y, Davis L (1996) Recognition of head gestures using hidden Markov models. In: Proceedings of 13th international conference on pattern recognition, pp 461–465

Nabiyouni M, Saktheeswaran A, Bowman DA, Karanth A (2015) Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. In: 2015 IEEE symposium on 3D user interfaces, pp 3–10

Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77:257–286

Robinett W, Holloway R (1992) Implementation of flying, scaling and grabbing in virtual worlds. In: Proceedings of the 1992 symposium on interactive 3D graphics, pp 189–192

Terven JR, Salas J, Raducanu B (2014) Robust head gestures recognition for assistive technology. In: Pattern recognition, pp 152–161

Wille M, Wischniewski S (2015) Influence of head mounted display hardware on performance and strain. In: Proceedings of the HFES annual meeting

Yan Z, Lindeman RW, Dey A (2016) Let your fingers do the walking: a unified approach for efficient short-, medium-, and long-distance travel in VR. In: 2016 IEEE symposium on 3D user interfaces (3DUI), pp 27–30

Zhao J, Allison RS (2017) Real-time head gesture recognition on head-mounted displays using cascaded hidden Markov models. In: 2017

IEEE international conference on systems, man, and cybernetics (SMC), pp 2361–2366
Zielasko D, Horn S, Freitag S, Weyers B, Kuhlen TW (2016) Evaluation of hands-free HMD-based navigation techniques for immersive data analysis. In: 2016 IEEE symposium on 3D user interfaces, pp 113–119