Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1056

ORIGINAL RESEARCH ARTICLE

VR based gesture elicitation for user—Interfaces with low vision

K. Kalaiselvi[1,*], R. Bhuvaneswari[2], R. Rekha[3], T. Rajesh Kumar[4], Rishav Banerjee[5], Omang Baheti[5]

[1] *Faculty of Engineering and Technology, Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur 603203, Chennai, Tamil Nadu, India*

[2] *Faculty of Engineering and Technology, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram 600089, Chennai, India*

[3] *Department of Computer Science and Business System, Panimalar Engineering College, Chennai 6000123, India*

[4] *Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, Tamil Nadu, India*

[5] *Department of Networking and communications, SRM Institute of Science and Technology, Kattankulathur 603203, Chennai, Tamil Nadu, India*

*** Corresponding author:** K. Kalaiselvi, mkkalai1981@gmail.com

ABSTRACT

User interfaces (UI) and menus in virtual reality (VR), which frequently replicate traditional UI for computers and smartphones, are not created factoring for individuals with low eyesight as they demand accurate pointing and good vision to engage effectively. As an alternative method of user interaction with UI, using gestures can be recommended. Comparing gesture-based interaction with the conventional point-and-click technique for changing system settings like volume, brightness, and window manipulation in order to test this hypothesis is employed. Accessibility, spatial awareness, and precision for those with low vision while lowering cognitive load and enhancing immersion for all users can be improved by leveraging gestures. The objective of the research work is to explore the framework of Gesture Elicitation in VR environments for users with low vision. In this research work the usage of gestures as a more effective and immersive means of interacting with menus, which will not only enhance the experience of normal VR users but also drastically reduce the friction experienced by those with visual impairments is proposed. User studies demonstrate a noticeable improvement in the aforementioned areas, with faster work completion times, more immersion, and better user satisfaction.

Keywords: virtual reality; gesture elicitation; accessibility

ARTICLE INFO

Received: 27 July 2023
Accepted: 14 December 2023
Available online: 5 March 2024

1. Introduction

The user interface (UI) in VR settings frequently uses the same concepts as their two-dimensional computing device counterparts. While the majority of interactive VR experiences make use of immersive settings and mirror real-world interactions, UI still uses the standard point-and-click method. Users are required to position their controllers at the appropriate buttons and fields on virtual panels, then use triggers or buttons to send inputs. Although this strategy works quite well for computers or even smartphones with touch inputs, it performs horribly in virtual reality. In addition to being a cumbersome method with many menus needing extremely precise movements, it lessens immersion by necessitating an interaction that is inappropriate for the medium.

The UI issues in VR is becoming more and more of a problem.

When individuals with low eyesight are taken into account, the problem of UI in VR becomes ever more challenging. The expectation that people who wear lenses wear VR headsets with glasses inside poses a first obstacle to comfort and immersion. Furthermore, persons with extremely weak vision hardly ever manage to fully traverse virtual worlds. It has taken a lot of effort to close this gap[1] and make VR more accessible to people with very weak vision. However, there hasn't been much done to make menus and user interfaces more accessible. Point and pinch controls are frequently used with hand tracking[2,3], although they have the same drawbacks as controllers in that they require accuracy and good vision to utilise.

Since its inception, virtual reality has relied on controllers as the most common schema for gathering user inputs, with alternative modes of interaction being primarily limited to research domains. Immersive VR experiences are now a reality that the typical user can accomplish thanks to the addition of hand posture estimation to the Oculus headset without the need for sophisticated sensors[4]. This has also made it possible to enhance UI interactions beyond how they are now.

To traverse UI and menus together, using gestures is suggested. This addresses the aforementioned issues in a number of ways. Gestures are used to improve the requirement for accuracy, boost interaction speed and efficiency, and offer a seamless, intuitive, and immersive way to traverse menus. By making use of gestures instead of a controller, also make it easier for people with low vision to execute, since performing an action is no longer linked with finding the right menu and selecting the right button, slider or field. There are two categories of gestures proposed, as follows:

- Quick Global Actions.
- Context Based Actions.

Speedy Worldwide Activities are planned to be general motions which can be executed from any climate. For instance, squeezing the ring finger to the thumb and hauling up or down increments or diminishes volume. Context put together activities with respect to the next hand are motions which act diversely contingent upon which application the client is presently centered around. This can be displayed by holding up the record and center finger to play or respite a melody, or a video relying on the application the client is in. The possible objective is to have the option to coordinate these into a significant headset provider structure at the working framework level.

HTC Vive and the PlayStation VR. Today, VR technology is used in a variety of applications, from gaming to education and training. The technology has come a long way since its early beginnings, and continues to evolve at a rapid pace. The development of VR applications has been driven by advances in computer technology, as well as by the increasing demand for immersive experiences in a wide range of industries.

- Advantages of a Raycast menu:
  a. Provides hands-free interaction.
  b. Ideal for selecting objects that are far away or hard to reach.
  c. Easy to implement and understand.
  d. Works well with the motion controllers.
- Drawbacks of this Raycast menu:
  a. May require more accuracy and precision to use.
  b. May feel less intuitive to some users.

The gestures can be classified few types.

1) System specific gestures:

These gesture types can be executed in any situation system wide. The purpose of these gestures is to allow for a quick and easy way to access OS features like system settings. These gestures should be quick to execute, designed to feel natural and not interfere the application specific gestures. One such example would be to change volume or brightness of the headset. The task is context independent, and can be accessed easily. Such a gesture will not detect the environment that it is being executed in and will perform consistently.

2) Application specific gestures:

These gesture types will behave differently depending on the active application. The purpose of these gestures is to combine activities with similar intent but in different context and settings to be executed in similar fashion. This would reduce the effort taken by the user to learn this system as it would require lesser time to imbibe these gestures into muscle memory.

1.1. Gesture elicitation

Gesture elicitation is a method used in the field of Human-Computer Interaction (HCI) to design intuitive gesture commands for controlling interactive devices, applications, and systems. The process involves observing and collecting data on the gestures made by end-users while performing various tasks, analyzing the collected data, and designing gesture commands that reflect end-users' behavior.

One of the benefits of gesture elicitation studies is that they provide valuable insights into how users interact with technology, which can inform the design of more user-friendly interfaces. For example, a study may reveal that users prefer certain types of gestures over others or that certain gestures are more intuitive for particular tasks.

Researchers have used gesture elicitation studies to investigate a variety of questions, such as the impact of increased production on gesture quality and whether increased production should be limited. They have also explored different aspects of gesture design, such as the refinement of promising gestures and the time it takes to find them.

To conduct a gesture elicitation study, researchers typically observe participants performing a task while recording their gestures. They then analyze the recorded data to identify common patterns and design gesture commands that reflect those patterns. Researchers may also use visualizations of the gesture data to help designers and developers better understand user behavior. The different kinds of gestures that can be elicited by the user in a convenient manner can be viewed.

One such example would be the play and pause action for different media applications. While a video player and a music player both contain the functionality for playing and pausing videos and music respectively, the act of playing and pausing remains common. Hence the same gesture can be overloaded for the aforementioned scenarios.

For our implementation, it is decided to set up a dummy VR environment with various menus displayed on a curved world space floating UI screen. The user will be tasked with performing various actions using both gestures and conventional analog motion controllers.

1.2. GE in XR

GE in extended reality (XR) applications. XR technologies, which include virtual reality (VR), augmented reality (AR), and mixed reality (MR), create immersive environments that blur the boundaries between the real and virtual worlds. GE plays an essential role in enhancing the user's interaction with XR systems by allowing users to use their body movements to control and manipulate digital objects.

In the context of XR, GE involves the use of sensors and cameras to capture the user's body movements and translate them into corresponding actions in the virtual environment. The use of GE in XR is significant because it allows users to interact with digital objects in a manner that is more natural and immersive than traditional input devices such as keyboards and mice.

One of the most important benefits of GE in XR is the ability to enhance the user's sense of presence in the virtual environment. By using their body movements to control and manipulate virtual objects, users can experience a more intuitive and immersive interaction with digital content. This enhanced sense of presence can lead to improved learning outcomes, increased engagement, and a more satisfying user experience.

Several techniques are used to elicit gestures in XR, including body tracking, hand tracking, and facial expression recognition. Body tracking involves the use of sensors to track the position and movement of the user's body in real-time. This technique is particularly useful for applications that require full-body movement, such as dance or sports simulations. Hand tracking, on the other hand, focuses on capturing the movement of the hands and fingers. This technique is commonly used in applications that require fine-grained control, such as drawing or sculpting. Facial expression recognition is a technique that captures the user's facial expressions and uses them to elicit gestures. This technique is useful for applications that require emotional expression, such as avatar-based communication systems.

2. Related work

2.1. Gesture elicitation

The exploration area of signal elicitation is as of now deep rooted, and various examinations have been directed in this field. In any case, a large portion of these examinations are restricted in scope, zeroing in essentially on planning frameworks for explicit undertakings or enhancing existing hand signal location frameworks. For example, Wu et al.[5] looked at how tabletop displays could get people to make gestures, while Sagayam and Hemanth[6] directed a review of various procedures utilized for recognizing hand stances.

For the purposes of this study, previous research in the field of gesture elicitation can be broadly divided into VR-related and non-VR-related work.

Gesture elicitation in desktop or other environments

Broad exploration has been led to investigate elective method for in-put for conventional gadgets determined to upgrade openness, vivid ness and natural use. Among the methods that have acquired broad reception are hand signals and movement controls. Specifically, multi-contact based hand signals are generally utilized in everyday gadgets while movement controls and body motions have turned into the standard in gaming consoles, like the Nintendo Wii and Nintendo Switch, and the Xbox Kinect, separately.

Analysts have investigated the execution of mid-air hand motions in PCs (laptops) to expand client input. Chua et al.[7], Chen et al.[8], and Matlani et al.[9] are notable examples of such studies. Using an ultrasonic sensor to detect mid-air gestures, Harrington et al.'s research[10] has shown that mid-air gestures reduce visual demand, shorten inter-action times, and improve task accuracy in an automotive setting. Gesture elicitation in virtual reality

The usage of mid-air hand signals in computer generated simulation (VR) has arisen as a promising area of examination, as it can possibly improve the client experience by giving a more natural and vivid connection point. Motion elicitation concentrates on zeroing in on mid-air signals in VR have essentially centered around specific undertakings that can be acted in VR conditions. For in-position, Leng et al.[11] showed the utilization of mid-air signals to collaborate with music in a virtual vivid climate. Additionally, Lin et al.[12] investigated different hand motions to control various articles in VR and examined the effect of stances on the related throughput. Yan et al.[13] showed a clever methodology for recovering items in VR utilizing hand signals like genuine getting. Besides, there have been concentrates on con-ducted to examine the utilization of hand signals for first-individual development in VR by Khundam[14].

In any case, there is a scarcity of studies that investigate the use of hand signals framework wide on a VR stage. This includes using context-aware gestures that perform various tasks with the same hand gesture depending on the context to navigate menus. Hence, further exploration is justified around here to decide the viability of framework wide hand motion use on improving the client experience in VR.

2.2. accessibility in VR

To work on the availability of computer generated reality (VR), a few scientists have investigated the incorporation of devices that improve the experience for clients with low vision. These devices comprise of an assortment of 15 utilities that help clients in deciphering the virtual climate, with a specific spotlight on those with very low vision. The devices incorporate text-to-discourse and article edge representation, which can support working on the translation of the virtual climate.

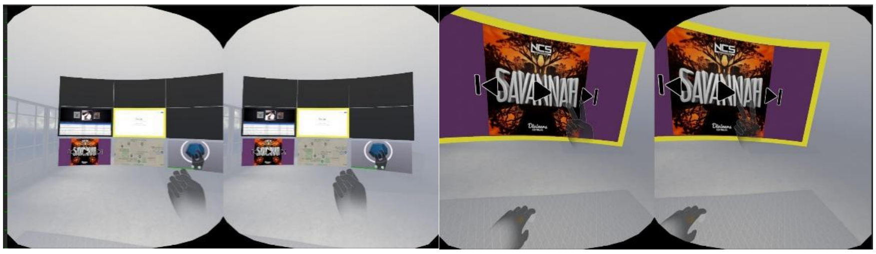
However, in the context of user interfaces (UI), the use of hand gestures has the potential to speed up the completion of various tasks. By restricting specific functionalities to hand signals, the need to peruse and between different UI components can be wiped out for explicit use cases, permitting the client to depend on muscle memory all things being equal. This approach can possibly improve availability in VR and could be investigated in future exploration.

3. Methods

3.1. Functionality

Functionality has been implemented to mimic the capabilities of a virtual operating system, akin to that of a conventional desktop or mobile operating system. The list of functionalities implemented, the hand pose and any additional associated gesture are as follows **Figure 1** showcase some of the tasks in the VR environment.

- Brightness up: left hand ring finger pinch, drag up.
- Brightness down: left hand ring finger pinch, drag down.
- Volume up: right hand ring finger pinch, drag up.
- Volume down: right hand ring finger pinch, drag down.
- Enable task manager: right hand close fist.
- Select window in task manager: right hand closed fist, drag around.
- Disable task manager: right hand open fist.
- Play and pause music: right hand index and middle finger separated (scissor).
- Play next song: right hand scissor, drag right.
- Play previous song: right hand scissor, drag left.
- Two dimensional scroll on a navigation map: right hand index pinch, drag around.
- Play and pause videos: right hand scissor.



(a) **(b)**

Figure 1. Tasks **(a)** task manager; **(b)** play and pause music.

3.2. Implementation

The system pipeline is as follows:

(1) The oculus interaction SDK for hand pose detection is used. This works by utilizing a scriptable object (SO) provided in the SDK as a means to store a target hand position. Each finger has various configuration possibilities. For instance, if wanted a finger gun gesture, the SO enumerable values are set as follows.
    (a) Thumb curl open.
    (b) Index flexion open.
    (c) Middle curl closed.
    (d) Ring curl closed.
    (e) Pinky curl closed.

(2) Once a target pose has been set, the SDK will send a callback upon the gesture being performed. The event is tracked, and a message is sent to an object with a system action handler interface, which performs the task associated with the hand pose.

(3) Additionally, the movements are tracked when the hand pose is active. This allows us to improve the intuitive nature of our system, since a hand pose can be connected to a group and use additional gestures for disambiguation. For example, a ring finger pinch can indicate the user wishes to alter brightness, and moving the hand up while performing the gesture would indicate increasing the brightness, while moving the hand down would indicate decreasing the brightness.

(4) Upon performing the gesture, the user sees the system respond appropriately.

4. User study

The user study has been structured to test for the following factors for both a conventional controller approach and using gestures:

- Intuitiveness.
- Ease of use.
- Immersion.
- Comfort.
- Fatigue.

The data points are scored on the basis of responses by participants filling out a questionnaire post the task execution. The list of tasks executed by the participants to assess their experience with this framework are as follows:

- Increase and decrease brightness.
- Increase and decrease volume.
- Switch between applications in the window manager.
- Play and pause music in the music player.
- Play and pause music in the video player.
- Drag a map around in the map viewer.

For the questionnaire, the users are first asked the following Yes or No questions:

- I have a visual impairment.
- I was able to perform most or all tasks with the controller.
- I was able to perform most or all tasks with gestures.

For the subjective analysis on the overall experience, the following questions were asked on a 5 point scale, with 1 correlating with highly disagree and 5 with highly agree. The questions have been framed such that higher scores imply gestures outperforming controllers in that respective category.

- Gestures felt more intuitive than controller actions.
- Gestures felt easier to execute than controller actions.
- Gestures felt easier to learn than controller actions.

- Gestures felt more precise than controller actions.
- Gestures felt more immersive than controller actions.
- Gestures felt less fatiguing than controller actions.

5. Results

Eleven participants were recommended with minor or major vision impairments. Most participants recruited for the user study had a reasonable vision im- perfection. From the 11 participants, 3 had near perfect vision, and so had a Gaussian blur filter applied to simulate poor vision. All users were first given a short tutorial on all the tasks, and their associated controller actions and gestures and then asked to replicate it three times. Every single applicant was able to execute every task in both control schemes.

The following table notes the feedback received from the various participants. The acronyms used are: P (participant), MIn (more intuitive), EE (easier to execute), EL (easier to learn), MP (more precise), MIm (more immersive) and LF (less fatiguing). As shown in **Table 1**. The responses also recorded in shown in **Figure 2a,2b,2c,2d,2e** and **2f**.

Table 1. Contains graphs visualizing and comparing the data recorded.

Participant survey results						
P	**MIn**	**EE**	**EL**	**MP**	**MIm**	**LF**
Id1	4	5	4	3	5	2
Id2	2	5	4	5	2	1
Id3	5	5	3	4	5	2
Id4	4	5	5	4	4	2
Id5	4	2	5	2	5	3
Id6	5	4	4	4	4	3
Id7	1	4	5	3	4	4
Id8	4	1	1	1	3	4
Id9	3	3	2	2	5	5
Id10	4	3	5	5	5	2
Id11	3	5	3	4	1	3

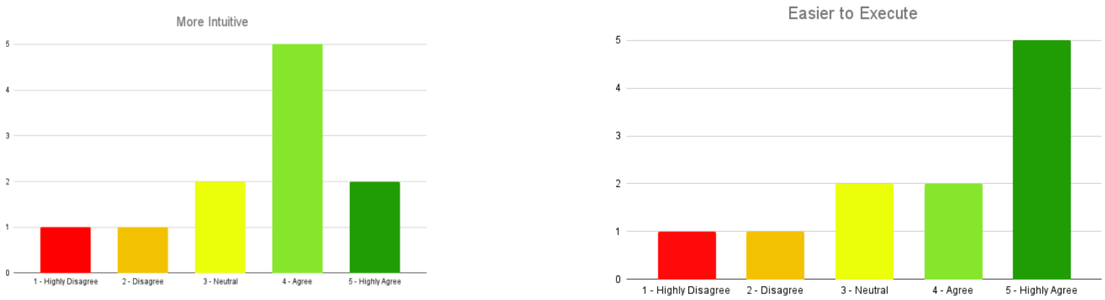


Figure 2a. More intuitive.

Figure 2b. Easy to record.

Figure 2. (*Continued*).

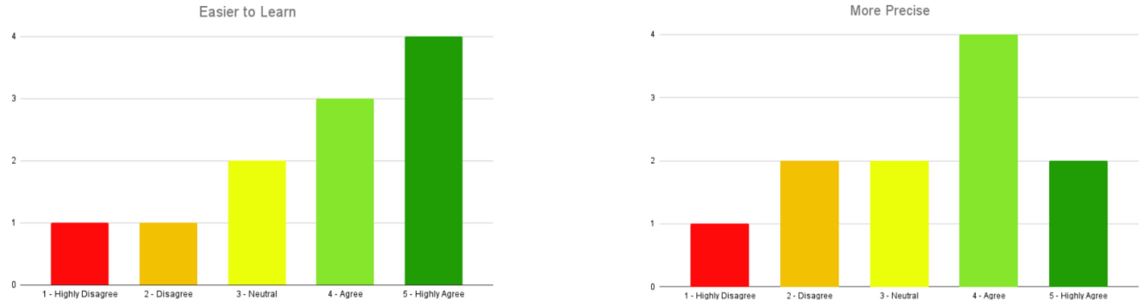


Figure 2c. Easy to learn. **Figure 2d.** Accuracy.

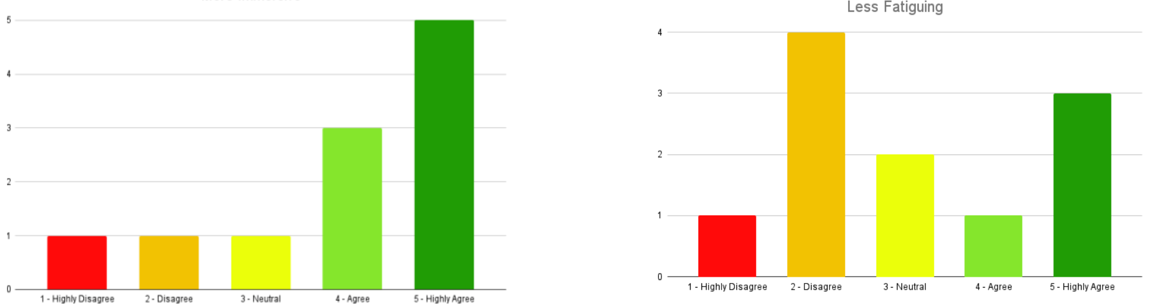


Figure 2e. More immersive. **Figure 2f.** Fatiguing.

Figure 2. Survey responses graphs.

6. Discussion

6.1. Notable observations

The average overall mean scored at 3.5, with the average median standing at 3.8, showing a clear preference towards gestures on the whole. The highest mean (3.9) and median (4) was recorded for MIm, whereas lowest mean (3.1) and median (3) was recorded for LF. Hand gestures are notably fatiguing given the need to hold your hand up and consistently perform gestures over an extended period of time. This is not an issue controllers face, since they can be pointed at the intended direction even with the arms relaxed.

6.2. Advantages

As hypothesised, using gestures seem to resonate with participants in improving immersion, precision, and ease of use. All participants notably performed and executed gestures with a single explanation, without requiring much clarification on why a certain gesture was to be executed a certain way. This can be attributed to the increasing research being done in the field of gesture elicitation, and what motions come naturally to humans collectively.

6.3. Drawbacks

A major drawback of this approach is overlap of certain gestures. In specific games or other VR experiences, developers might not be aware of the existence of global actions. There are two potential situations that may arise, and suggestions corresponding solutions:

- Using context gestures without the presence of said context: For example, using the context gesture to control various multimedia settings, without any multimedia being present. In this case there is no actual overlap.
- Global action overlap, or context action overlap: If there is a direct overlap, the developer can be intimated during development of such a conflict. Should it be ignored however, the system action can be temporarily disabled.

The general expectation is to find a solution with further studies rather than the above proposed workarounds.

This entire system was built on the oculus hand pose detection SDK, which is not consistent as a framework. 3 participants noted the gesture detection software seemed rather “glitchy” and unreliable, which is an existing quirk of the oculus hand pose detection SDK. It requires good lighting and very clear line of sight from the headset’s sensors to detect the various gestures made, as it uses a pose estimation deep learning algorithm (this algorithm is proprietary, and hence it is not possible to modify or tweak its performance). Since this is primarily driven by machine learning, the expectation is it will get better with time and improved data, while fine tuning the architecture to run faster and with improved accuracy.

7. Conclusion

The current UI paradigms in VR rely heavily on controllers, which can be problematic for users with poor vision or those who struggle with precise movements.

The study proposes the use of hand gestures as an alternative means of interacting with VR menus and UI. Two categories of gestures, Quick Global Actions and Context-Based Actions, were identified and found to be more intuitive and accessible than complex button combinations.

However, prolonged use of hand gestures can cause fatigue in the arms, and implementing them across different VR headsets can be challenging due to inconsistencies in hand pose detection algorithms.

To address these issues, developers could establish a consistent design language for hand gesture interaction and improve the accuracy of hand pose detection algorithms. Despite these challenges, the use of hand gestures in VR has clear advantages, such as improving immersion, precision, and ease of use.

Developers must carefully consider the reliability and accuracy of gesture detection software to ensure a positive user experience. Overall, this study can serve as a reference for rethinking UI interactions in VR and guiding future research in this area.

Author contributions

Conceptualization, KK and RB (R. Bhuvaneswari); methodology, KK; software, RB (R. Bhuvaneswari); validation, OB, RR and TRK; formal analysis, RB (R. Bhuvaneswari); investigation, RR; resources, OB; data curation, RB (Rishav Banerjee); writing—original draft preparation, RR; writing—review and editing, TRK; visualization, KK; supervision, RB (Rishav Banerjee); project administration, RR; funding acquisition, OB. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Zhao Y, Cutrell E, Holz C, et al. Seeing VR: A set of tools to make virtual reality more accessible to people with low vision. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Published online May 2, 2019. doi: 10.1145/3290605.3300341
2. Oculus. Available online: https://www.meta.com/ help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking-quest-2/ (accessed on 20 June 2020).
3. Immersed, Virtual reality workspace, Available online: https://immersed.com/ (accessed on 20 June 2020).
4. Oculus. Hand pose detection interaction software development kit. Available online: https://developer.oculus.com/documentation/unity/ unity-isdk-hand-pose-detection (accessed on 20 June 2020).
5. Wu M, Balakrishnan R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Proceedings of the 16th annual ACM symposium on User interface software and technology. Published online November 2, 2003. doi: 10.1145/964696.964718
6. Sagayam KM, Hemanth DJ. Hand posture and gesture recognition techniques for virtual reality applications: a survey. Virtual Reality. 2016, 21(2): 91-107. doi: 10.1007/s10055-016-0301-0
7. Chua SND, Chin KYR, Lim SF, et al. Hand Gesture Control for Human–Computer Interaction with Deep

Learning. Journal of Electrical Engineering & Technology. 2022, 17(3): 1961-1970. doi: 10.1007/s42835-021-00972-6

8. Chen LC, Cheng YM, Chu PY, et al. The Common Characteristics of User-Defined and Mid-Air Gestures for Rotating 3D Digital Contents. Universal Access in Human-Computer Interaction Interaction Techniques and Environments. Published online 2016: 15-22. doi: 10.1007/978-3-319-40244-4_2
9. Matlani R, Dadlani R, Dumbre S, et al. Virtual Mouse using Hand Gestures. 2021 International Conference on Technological Advancements and Innovations (ICTAI). Published online November 10, 2021. doi: 10.1109/ictai53825.2021.9673251
10. Harrington K, Large DR, Burnett G, et al. Exploring the Use of Mid-Air Ultrasonic Feedback to Enhance Automotive User Interfaces. Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. Published online September 23, 2018. doi: 10.1145/3239060.3239089
11. Leng HY, Norowi NM, Jantan AH. A User-Defined Gesture Set for Music Interaction in Immersive Virtual Environment. Proceedings of the 3rd International Conference on Human-Computer Interaction and User Experience in Indonesia. Published online April 18, 2017. doi: 10.1145/3077343.3077348
12. Lin W, Du L, Harris-Adamson C, et al. Design of Hand Gestures for Manipulating Objects in Virtual Reality. Lecture Notes in Computer Science. Published online 2017: 584-592. doi: 10.1007/978-3-319-58071-5_44
13. Yan Y, Yu C, Ma X, et al. VirtualGrasp. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Published online April 19, 2018. doi: 10.1145/3173574.3173652
14. Khundam C. First person movement control with palm normal and hand gesture interaction in virtual reality. 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE). Published online July 2015. doi: 10.1109/jcsse.2015.7219818