

Original Article

# Implementation of Immersive Virtual Hands using Virtual Reality System Controllers

Arman A. Hovhannisyan

Information Security and Software Engineering, National Polytechnic University of Armenia, Yerevan, Armenia.

Corresponding Author: [aahovhannisyan1@gmail.com](mailto:aahovhannisyan1@gmail.com)

Received: 26 February 2023

Revised: 12 May 2023

Accepted: 23 May 2023

Published: 25 June 2023

**Abstract** - Virtual reality is a technology that makes computer-generated objects and environments appear natural. Immersive virtual reality aims to involve the user in the virtual environment at the most, giving the user the feeling that they are actually inside the virtual world. Immersion is the user's engagement with a virtual reality system that results in being in a flow state. Presence within the context of virtual reality is a phenomenon that allows users to interact naturally with a virtual world through technology. For a virtual environment to be realistic, it must provide a high level of immersion and presence. One way to achieve this is by rendering the user's body in the virtual environment. Particularly, rendering the user's hands makes it possible to simulate user-object interactions naturally as they appear in real life. Despite the high volume of research efforts made towards increasing the feeling of immersion and presence in virtual worlds, the task of providing users with more realism is still actual. This paper proposes a method for simulating hand gestures as they appear in real life, using virtual reality system controllers. As a result, it gives users the feeling of natural movements and actions within the virtual environment. The proposed method commits to increasing immersion and presence in virtual reality environments.

**Keywords** - Hand gesture, Human-computer interaction, Immersion, Presence, Virtual reality.

## 1. Introduction

Virtual reality (VR) is a simulated experience that provides users with immersion through virtual reality equipment and pose-tracking mechanisms. The equipment, commonly known as a VR headset, allows users to navigate the virtual world, look around and interact with virtual objects. VR headsets typically consist of a head-mounted device with small near-eye displays, controllers to interact with the virtual world, and a pose-tracking mechanism.[1] Although VR is a novel interactive interface for human-computer interaction (HCI), it has applications in various industries, including education, medicine, and entertainment. The majority of HCI research focuses on making computer interfaces more usable to increase human-computer interaction, making it actual for VR applications to think about developing an improved manner of interaction. VR applications comprise two types of components: environments and user interfaces (UI). The environment is the world the user appears in after wearing the headset, and UI is the set of elements the user interacts with to navigate the environment. When designing a VR application, special attention should be paid to implementing the UI to avoid user frustration. More than one-third of the users interrupted their experience during the market research because they did not understand the program operation, according to Yue.[2] To have a user-friendly UI, the application must be designed to

be interactive and immersive. Immersion allows users to feel like they are acting in the real world.[3] As summarized by Coomans and Timmermans,[4], interactivity must have the following characteristics:

- Sensitivity to natural commands: the system must understand actions that appear natural to the user.
- Real-time feedback: a user's action must immediately be reacted upon.
- Natural feedback: the system must respond in a way that appears natural to the user.

Users' reports of a sensation of presence in virtual environments (VEs) have frequently been used to determine their efficacy. Presence is defined as the subjective experience of being in one place or environment, even when one is physically situated in another.[5] Rendering the user's body in virtual reality increases immersion and presence. Displaying the user's hand as the primary body part for interaction enables natural user interactions with the virtual world.[6] To investigate the virtual hand illusion (VHI), Lin and Jörg conducted two studies in which participants took on controllable hand models with six distinct appearances: realistic hand, toony hand, very toony hand, zombie hand, robot hand, and wooden block. The results show that even though all the models may be considered for creating the



illusion of hand interaction, the effect of involvement is low for non-realistic models and high for anthropomorphic models.[7] In another research, Argelaguet, Hoyet, Trico, and Lécuyer undertook a study to examine the experience of the embodiment when engaging with three diverse virtual hand representations, each of which offered a different level of visual realism while maintaining the same precise control mechanism. The findings reveal that non-realistic hand models have a weaker sense of ownership, and there is also more mismatch between the user's actions in real life and virtual hand model animations. These parameters are improved for the realistic hand model with direct mapping of degrees of freedom (DOF) between real and virtual hands.[8] It can be concluded that for a VR application to have an increased sense of immersion, there is a need for a visually realistic virtual hand model with life-like hand gestures and motions.

Prior research exists on virtual hand modeling, introducing various virtual hand models. E. Peña-Pitarch, N. Falguera, and J. Yang present a 25 DOFs hand model. The model is developed based on hand anatomy kinematics. Due to the high number of DOFs, the Denavit-Hartenberg convention is employed to define joint relationships.[9] Another method is suggested by H. Wan, Y. Luo, S. Gao, and Q. Peng, where the metaball modeling method is used to create the virtual hand. To improve the visual appeal, texture mapping is applied to the model. The model consists of 3 layers (skeleton, muscle, and skin layers) to enhance the realistic motion and deformation of the hand. [10] Finally, L. Lin et al. discusses how differences in size between a user's real hand and the virtual hand can affect the perception of the virtual hand illusion and user experience. In their research, they use three levels of robotic hand scaling.[32] Although these models provide immersion and presence to some extent, there is still a place for improvement to get a more visually realistic hand representation. Furthermore, the existing work does not consider animating hand gestures as they appear in real life.

Despite the fact that much research effort has been made to increase the feeling of immersion in a virtual environment, the task of providing users with more realism and immersion is still actual. An essential factor to consider in this context is the hand-tracking method. Much research on this topic concentrates on haptic feedback and motion-based interaction through "hand-held" devices (e.g., haptic gloves).[12-18] However, these devices cannot be commercialized for the public because of their limited and costly usage.[33] Another technique used for hand tracking is the computer vision-based approach, which implies utilizing high-resolution depth cameras to capture the bare hands, and then using this information to position the virtual hand in the exact location and posture. Leap Motion Controller, [20] used for tracking hand position, is known to have gesture recognition issues. It is a hand-tracking device

being mounted on the HMD, which means that it has a single point of view of the user's hands. Tracking issues come over when the user's fingers are covered by his palm. The tracking system tries to make assumptions about the covered finger positions by analyzing other hand parameters. However, it still results in many inaccurate positions and gestures, leading to poor user experience and frustration.[21] Thus, this work is focused on employing VR headset controllers as an input method rather than haptic devices and computer vision-based techniques. Recent research has also focused on multimodality - the combination of multiple communicative modes, which improves immersion by interacting with two and more senses. This topic still needs more investigation because of the lack of simple systems and equipment capable of detecting senses like smell and taste.[22]

This paper proposes a method for simulating hand gestures as they appear in real life. A novel method is proposed for mapping headset controller buttons to hand gestures in the virtual environment. Animation state machines are defined to enhance smooth animations between hand poses.

## 2. Materials and Methods

A virtual hand model is needed to simulate the hand gestures in a VE. For that purpose, the hand kinematic model is considered, and a 3D model is adopted based on the following criteria: the model must provide all the needed degrees of freedom of the hand joints, and it must be visually realistic enough for the users to think of it as their own. Afterwards, the HTC Vive Cosmos Elite VR system controllers are reviewed, and the controller buttons' press/release events are associated with specific hand gestures based on the natural position of the controller in hand.

### 2.1. Hand Model

The human hand is a complex organ made of bones, muscles, blood vessels, etc. The virtual hands have to behave as in real life to achieve a high level of immersion. For that purpose, first, the biomechanics of the hand is considered. Figure 1 illustrates the kinematic model of the hand, where joints and bones are articulated by 27 degrees of freedom (DOF).[24, 34] To provide users with a life-like experience, the virtual model has to be realistic enough and contain all the joints with corresponding DOF to simulate hand gestures. Brief research showed that the VR Hands and FP Arms Pack Unity package models meet these requirements.[25] Assets of this package use custom and fast standard skin shader, which supports features like translucency, tint, rim light, and ramp.[26] The model used in this simulation theoretically provides 48 DOF (Figure 2(a)), of which 21 are used for the animations throughout this paper. The mesh used to define the 3-D shape of this model is made of 5414 triangles and 2997 vertices (Figure 2(b)), thus providing high-fidelity hand representation.

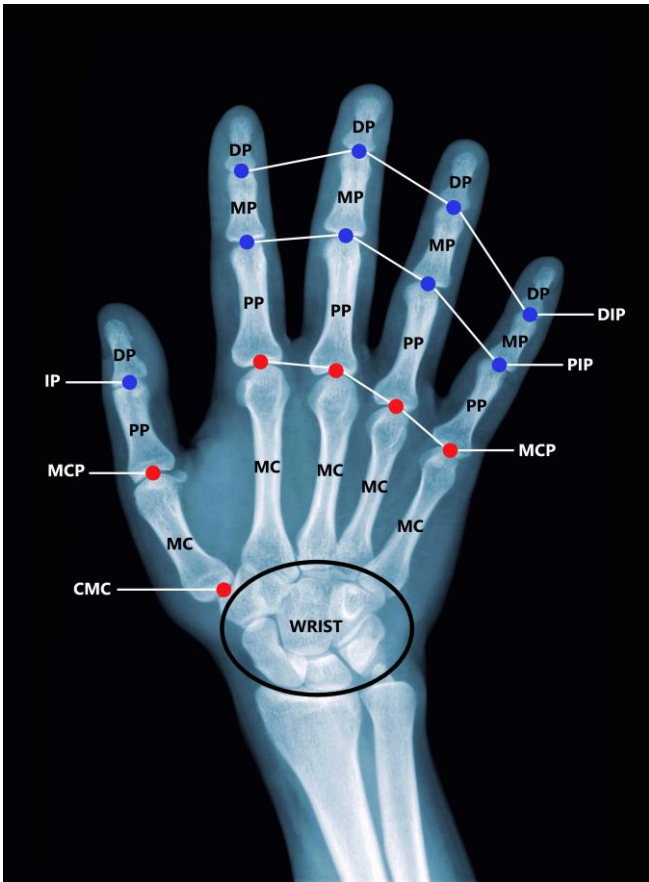


Fig. 1 Kinematic model of the human hand. DP - Distal Phalanges. MP - Middle Phalanges. PP - Proximal Phalanges. MC - Metacarpal bones. DIP - Distal interphalangeal joint. PIP - Proximal interphalangeal joint. MCP - Metacarpophalangeal joint. IP - Interphalangeal joint of the thumb. CMC - Carpometacarpal joint. The blue points represent 1 DOF joint. The red points represent 2 DOF joints. The wrist has 6 DOF.

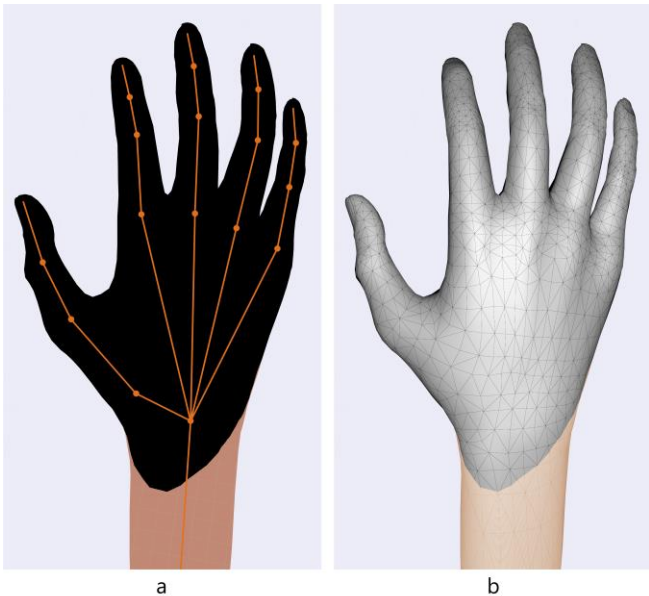


Fig. 2 Hand 3D model used in the simulation. (a) The model skeletal structure. (b) The model mesh structure.

**2.2. VR Controllers**

The VR equipment selected for simulating scenes throughout this paper is the HTC Vive Cosmos Elite virtual reality headset. It is a PC-tethered platform consisting of a head-mounted device (HMD), two base station units, and two wand-like wireless motion controllers. The base station units are set up for user pose tracking, and each is equipped with infrared lasers as well as LED lights. The lasers rotate and scan the area to be tracked with fixed time intervals. Photo sensors built into the HMD and controllers pick up the LED lights emitted from the base station units and establish the relative position of the objects in 3D space.[35]

The controllers (Figure 3) are used to interact with the VR world. Twenty-four infrared sensors are distributed throughout the controller's ring using the base stations to locate the controller. The SteamVR Tracking is used as a hardware/software solution to track the controller locations. Having update rates within the range of 250 Hz – 1 kHz, it follows the controller location with an accuracy of a millimeter.

For the users to interact with the virtual world through VR controllers, it is not mandatory that the hand model is represented in the VE, nor is it required for the model to perform smooth transitions between different hand poses: a ray interactor may be used instead to control objects on distance. However, as described earlier, rendering the user's hand in VE increases immersion. For the hand model to be authentic, it must also articulate in a certain way in response to input actions. E.g., when the grip button is pressed on the controller to grasp an object in the VE, the virtual hand fingers must be closed. Below are the proposed rules for changing the hand gesture based on input action.

The trigger button is usually used to select or click some objects (prevalently menu items). In real life, such an action usually would be performed with the index finger open and the other fingers closed. Thus, it is proposed that only the index finger is open when pressing the trigger button. This decision is also due to the fact that when holding the controller in its natural position, the index finger is on the trigger button. The open state for the other four fingers is determined on the trackpad clockwise: thumb - the left quarter, middle finger - the top quarter, ring finger - the right quarter, little finger - the bottom quarter. The grip button is used to grasp objects, so to provide users with the natural feeling of holding the object, it is proposed that all the fingers must be closed when pressing the grip button.

**3. Implementation**

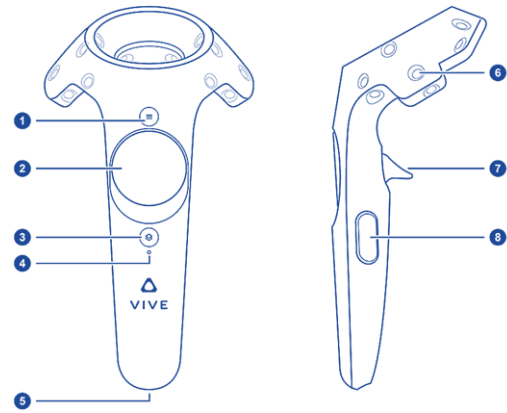
First, animations are implemented to provide the user with a smooth experience of flexing/extending the fingers on the corresponding button press actions. Afterwards, these animations are linked to the button press/release events. The

simulations throughout this paper are conducted within the Unity Engine, a cross-platform real-time development engine supporting different desktop, mobile and virtual reality platforms. The engine may be used to build three-dimensional scenes and interactive simulations. It also provides a mechanism to define object animations. [28]

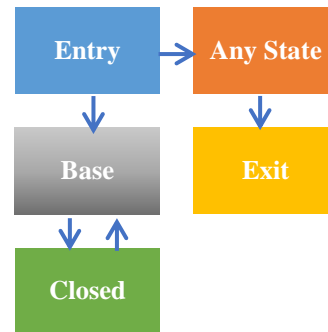
**3.1. Animation**

Animations in the Unity Engine are created using an Animator Controller. An animated game object's logic is managed by a Unity item called an Animator Controller. The Animator Controller consists of States and Sub-State Machines. Transitions define the connections between states. States represent the object positions, and transitions run the Animation Clips whenever a state change is performed. The Animation Machine states are shown in Figure 4, where Base represents the idle state (no button is pressed, all fingers are closed according to the button pressed), and Closed is the state where some fingers are closed according to the button pressed. The Entry state defines which state the machine starts, which is the Base state in our case. As seen from the figure, there are two transitions between Base and Closed states. The Base to Close transition is for the button press event, and the Close to Base transition is for the button release event. The State Machine also defines two additional states, Any State and Exit. Any State is a special state for making transitions to a specific state regardless of which state the machine is in currently. This state is left unused because it is always known which states the transitions should occur between (Base to Closed whenever a button press event occurs and Closed to Base whenever a button release event occurs). The Exit state allows the machine to finish animations for the current scene. Because the animations must run throughout the simulation, and there is no case when an interaction event will not trigger a specific animation, the Exit state is also left unused.

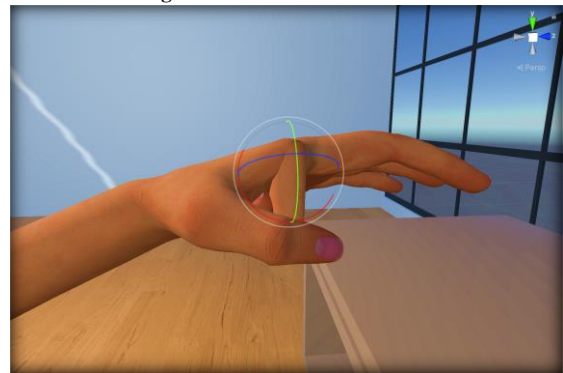
Each state machine has a conditional variable linked to itself, which controls whether the transition should occur at a given instant. The state machine transitions occur during the time phase when the conditional variable linked to this machine is in between some pre-defined values. More on conditional variables and pre-defined values are discussed in the next section. The implementation considers one state machine for each interaction event, resulting in overall six state machines. The interaction events are as follows: the trigger button press/release event (animating all the fingers except the index finger), the grip button press/release event (animating all the fingers), and one interaction event per trackpad four quarters (animating all the fingers except the finger defined for the specific quarter). The Closed state for each machine is achieved by rotating the finger phalanges to the needed degree to get the natural closed look of the hand. Figure 5 demonstrates the index finger closed state, and Figure 6 demonstrates the fist - all the fingers in the closed state.



Source: [29]  
**Fig. 3 HTC Vive controllers. (1) Menu button, (2) Trackpad, (3) System button, (4) Status light, (5) Micro-USB port, (6) Tracking Sensor, (7) Trigger, (8) Grip button.**



**Fig. 4 Animation State Machine**



**Fig. 5 Index finger in the closed state**



**Fig. 6 Hand in the closed state**

### 3.2. Handling Controller Inputs

As discussed earlier, animations must occur when pressing (releasing) controller buttons, specifically the trigger button, grip button, and trackpad four quarters. Thus, there is a need to link these events with the animation state machine. For this reason, a script in C# language is created to listen to the controller events and change the state machine conditional variable value accordingly.

To perform the animation smoothly, it is necessary to have the interval in seconds from the last frame to the current one at any given moment. It is determined in Equation 1, where  $t_2$  is the current time in seconds, and  $t_1$  is the time of the last frame.

$$\partial t = t_2 - t_1 \quad (1)$$

To control the animation speed, a new environment variable is declared. Denoting it by letter  $s$ , the maximal difference of the conditional variable value during one frame (denoted as  $m$ ) can be determined as follows:

$$m = s * \partial t \quad (2)$$

The state machine conditional variable value is expressed as Equation 3, where  $c$  is the conditional variable,  $t$  is the target value for the conditional variable value to reach and transition to be finished, and  $f$  is a function moving the value of the conditional variable towards the target value with steps not increasing  $m$ .

$$c = f(c, t, m) \quad (3)$$

According to Bodleian Libraries,[36] the recommended frames per second (FPS) for virtual reality systems is 90. Lower values will likely cause nausea, disorientation, and general discomfort that might lead to motion sickness.

Considering 90 as the minimal FPS, it is easy to calculate that the minimal time lapse between 2 subsequent frames is 0.01 seconds. Because Equations 1, 2, and 3 demonstrate that the conditional variable value is dependent on the interval in seconds from the last frame to the current one, the following can be concluded:

- $c > 0.01$ . The button is pressed, and a state machine transition from the Base state to the Closed state must be performed.
- $c < 0.01$ . The button is released, and a state machine transition from the Closed state to the Base state must be performed.

Because the controller buttons have two states – pressed and released, the button states in the program are represented as Boolean values. Based on these facts, the state machine conditional variable is set to float between 0.01 and 1.

The implementation is available at <https://github.com/aahovhannisyanyan/vr-immersive-hands>.

## 4. Conclusion

This paper proposed a method for simulating hand gestures as they appear in real life to make the virtual environment more immersive and enhance the feeling of presence for the user. Although there are techniques to track the bare hand gestures to provide more immersive experiences, it was decided to use the controllers included with all HMD devices because of the existing issues with the haptic gloves and computer vision-based techniques. The HTC Vive Cosmos Elite virtual reality headset controllers served as a user input method. The controller buttons were mapped to specific actions in the virtual environment based on their natural position in the hand. Animations were made for smooth, life-like transitions.

## References

- [1] Wikipedia, Virtual Reality. [Online]. Available: [https://en.wikipedia.org/wiki/Virtual\\_reality](https://en.wikipedia.org/wiki/Virtual_reality)
- [2] G. Yue, “3D User Interface in Virtual Reality,” in *HCI 2021*, vol. 1498, pp. 418-423, 2021. [CrossRef] [Google Scholar] [Publisher Link]
- [3] M. Slater, and A. Steed, “A Virtual Presence Counter,” *IEEE Presence*, vol. 9, no. 5, pp. 413-434, 2000. [CrossRef] [Google Scholar] [Publisher Link]
- [4] M.K.D. Coomans, and H.J.P. Timmermans, “Towards a Taxonomy of Virtual Reality user Interfaces,” *Proceedings 1997 IEEE Conference on Information Visualization*, 1997. [CrossRef] [Google Scholar] [Publisher Link]
- [5] B. Witmer, and M. Singer, “Measuring Presence in Virtual Environments: A Presence Questionnaire,” *MIT Presence: Teleoperators and Virtual Environments*, vol. 7, no. 3, pp. 225-240, 1998. [CrossRef] [Google Scholar] [Publisher Link]
- [6] V. Schwind et al., “These are not My Hands!”: Effect of Gender on the Perception of Avatar Hands in Virtual Reality,” *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1577-1582, 2017. [CrossRef] [Google Scholar] [Publisher Link]
- [7] L. Lin, and S. Jörg, “Need a Hand?: How Appearance Affects the Virtual Hand Illusion,” *Proceedings of the ACM Symposium on Applied Perception*, pp. 69-76, 2016. [CrossRef] [Google Scholar] [Publisher Link]
- [8] F. Argelaguet et al., “The Role of Interaction in Virtual Embodiment: Effects of the Virtual haNd Representation,” *2016 IEEE Virtual Reality (VR)*, 2016. [CrossRef] [Google Scholar] [Publisher Link]

- [9] E. Peña-Pitarch, N.T. Falguera, and J. Yang, “Virtual Human Hand: Model and Kinematics,” *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, no. 5, pp. 568-579, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] H. Wan et al., “Realistic Virtual Hand Modeling with Applications for Virtual Grasping,” *VRCAI '04*, pp. 81–87, 2004. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] E. Sudhakaran Rao, “Automatic Control of Wheel Chair using Hand Gestures in Signal Processing,” *SSRG International Journal of VLSI & Signal Processing*, vol. 1, no. 1, pp. 13-15, 2014. [[CrossRef](#)] [[Publisher Link](#)]
- [12] Mingyu Kim et al., “Grasping VR: Presence of Pseudo-Haptic Interface Based Portable Hand Grip System in Immersive Virtual Reality,” *International Journal of Human-Computer Interaction*, vol. 36, no. 7, pp. 685-698, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] J. Hwang, J. Jung, and G. Kim, “Hand-held Virtual Reality: A Feasibility Study,” *VRST '06: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 356-363, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] J. Perret, and E.V. Poorten, “Touching Virtual Reality: A Review of Haptic Gloves,” *ACTUATOR 2018: 16<sup>th</sup> International Conference on New Actuators*, 2018. [[Publisher Link](#)]
- [15] Jonathan Blake, and Hakan B. Gurocak, “Haptic Glove with MR Brakes for Virtual Reality,” *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 5, pp. 606-615, 2009. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] V. Popescu, G. Burdea, and M. Bouzit, “Virtual Reality Simulation Modeling for a Haptic Glove,” *Proceedings Computer Animation 1999*, 1999. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Vu Trieu Minh, Nikita Katushin, and John Pumwa, “Motion Tracking Glove for Augmented Reality and Virtual Reality,” *Journal of Behavioral Robotics*, vol. 10, no. 1, pp.160-166, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] C. Våpenstad et al., “Perceiving Haptic Feedback in Virtual Reality Simulators,” *Surgical Endoscopy*, vol. 27, pp. 2391-2397, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] P. Dhivya Bharathy et al., “Hand Gesture Recognition for Physical Impairment Peoples,” *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 10, pp. 6-10, 2017. [[CrossRef](#)] [[Publisher Link](#)]
- [20] UltraLeap, Leap Motion Controller Data Sheet. [Online]. Available: [https://www.ultraleap.com/datasheets/Leap\\_Motion\\_Controller\\_Datasheet.pdf](https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf)
- [21] Otso Pietikäinen et al., “VRChem: A Virtual Reality Molecular Builder,” *MDPI Applied Sciences*, vol. 11, no. 22, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Mingyu Kim, Changyu Jeon, and Jinmo Kim, “A Study on Immersion and Presence of a Portable Hand Haptic System for Immersive Virtual Reality,” *MDPI Sensors*, vol. 17, no. 5, p. 1141, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Ramya Ravibarathi, and Tamilarasu Viswanathan, “Assimilation of Gesture using 9 Axis Accelerometer Sensor,” *SSRG International Journal of Electrical and Electronics Engineering*, vol. 5, no. 2, pp. 1-4, 2018. [[CrossRef](#)] [[Publisher Link](#)]
- [24] George ElKoura, and Karan Singh, “Handrix: Animating the Human Hand,” *SCA '03*, pp. 110-119, 2003. [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Unity Asset Store Website, VR Hands and FP Arms Pack. [Online]. Available: <https://assetstore.unity.com/packages/3d/characters/humanoids/vr-hands-and-fp-arms-pack-77815>
- [26] Unity Documentation website. [Online]. Available: <https://docs.unity3d.com/Manual/shader-introduction.html>
- [27] R. Jayasree et al., “Gesture Based Assistance System for Aphasia Patient,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 5, no. 7, pp. 28-33, 2018. [[CrossRef](#)] [[Publisher Link](#)]
- [28] Wikipedia, Unity (Game Engine). [Online]. Available: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [29] Vive website, About the VIVE Controllers. [Online]. Available: [https://www.vive.com/us/support/vive/category\\_howto/about-the-controllers.html](https://www.vive.com/us/support/vive/category_howto/about-the-controllers.html)
- [30] Dhaval Waghulde et al., “Wireless Cursor Control by Gesture Recognition,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 2, no. 11, pp. 37-41, 2015. [[CrossRef](#)] [[Publisher Link](#)]
- [31] Hoang Thi Phuong, “Researching Robot Arm Control System Based On Computer Vision Application And Artificial Intelligence Technology,” *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 1, pp. 24-29, 2021. [[CrossRef](#)] [[Publisher Link](#)]
- [32] Lorraine Lin et al., “The Effect of Hand Size and Interaction Modality on the Virtual Hand Illusion,” *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [33] Mandayam A. Srinivasan, and Cagatay Basdogan, “Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges,” *Elsevier Computers & Graphics*, vol. 21, no. 4, pp. 393-404, 1997. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] M. Pouliquen et al., “Virtual Hands and Virtual Reality Multimodal Platform to Design Safer Industrial Systems,” *Elsevier Computers in Industry*, vol. 58, no. 1, pp. 46–56, 2007. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [35] Wikipedia, HTC Vive. [Online]. Available: [https://en.wikipedia.org/wiki/HTC\\_Vive](https://en.wikipedia.org/wiki/HTC_Vive)
- [36] Bodleian Libraries, Virtual Reality: How to use VR Headsets. [Online]. Available: <https://libguides.bodleian.ox.ac.uk/vr/headsetsvr>