



A Preliminary Investigation on a Multimodal Controller and Freehand Based Interaction in Virtual Reality

Nicola Capece^(✉), Monica Grusso, Ugo Erra,
Rosario Catena, and Gilda Manfredi

Dipartimento di Matematica, Informatica ed Economia,
Università degli Studi della Basilicata, Potenza, Italy
{nicola.capece,monica.grusso,ugo.erra}@unibas.it
{rosario.catena,gilda.manfredi}@studenti.unibas.it

Abstract. In the last years, the synergy between VR and HCI has dramatically increased the user's feeling of immersion within virtual scenes, improving VR applications' user experience and usability. Two main aspects have emerged with the evolution of these technologies concerning immersion in the 3D scene: locomotion within the 3D scene and interaction with the components of the 3D scene. Locomotion with classical freehand approaches based on hand tracking can be stressful for the user due to the need to keep the hand still in a specific position for a long time to activate the locomotion gesture. Likewise, using a classic Head Mounted Display (HMD) controller for the interaction with the 3D scene components could be unnatural for the user, using the hand to pinch and grab the 3D objects. This paper proposes a multimodal approach, mixing the Leap Motion and 6-DOF controller to navigate and interact with the 3D scene to reduce the locomotion gesture stress based on the hand tracking and increase the immersion and interaction feeling using freehand to interact with the 3D scene objects.

Keywords: User interaction · Locomotion · HCI · Virtual reality · Visualization · Freehand · 3D graph

1 Introduction

Today, one of the main challenges concerns the development of new approaches to interact with 3D environments in Virtual Reality that overcome the traditional techniques based, for example, on controllers [17,18]. For this reason, more and more approaches have been investigated in this direction [21,22] in addition to new methods and studies of locomotion [3,19,23] to increase the level of user immersion in the virtual environment, trying to reduce as much as possible the interference feelings of the real environment. Freehand through hand tracking indeed represents one of the innovations that in recent years is

becoming increasingly popular in the field of human-computer interaction applications [5] and VR [1, 16, 27]. Indeed, several new technologies and devices such as omnidirectional treadmill (ODT) [12] and integrated hand tracking [28] with HMD such as Oculus Quest [10] are emerging to increase the user’s feeling of immersion in the virtual scene. However, some of them, such as ODT, also need very efficient support technologies, such as harness and haptic devices, allowing users to achieve a sufficiently comfortable and low-stress experience.

One of the main limitations of the hands tracking and gesture recognition is the muscle fatigue and tired arm, also called the gorilla-arm effect, due to keeping for long time hands and fingers visible to the tracking devices [8], such as Depth Cameras, Leap Motion, Webcam, and so on. In our previous study [3], we analyzed the different type of freehand-steering locomotion techniques by providing an empirical evaluation and comparison with the use of HTC Vive controllers. One of the issues emerging from this study is that fatigue can affect the user when a hand gesture must be reproduced continuously to perform the movement.

Although virtual object interaction through hand gestures certainly increases the users’ immersion feeling in the virtual scene, the use of freehand for the locomotion forces them to keep the arm still for a long time, consequently increasing fatigue and stress. [29]. For this reason, we introduced a multimodal approach allowing users to move through the virtual scene with a 6-DOF controller and contextually interact with the virtual object using well-defined hand gestures detected through the Leap Motion controller. To the best of our knowledge, it is the first approach that combines natural user hand interaction and navigation through a controller. A 3D graph visualization in VR, where the user can explore and interact with a 3D graph and its nodes, was used as a case study for our multimodal approach.

The remainder of this paper is structured as follow: some related work is reported in Sect. 2; some implementation details and used developing tools are reported in Sect. 3; Sect. 4 shows our proposed system and its architecture; Sect. 5 shows the case study dealt, and finally Sect. 6 reports the conclusions and future works.

2 Related Work

Performing locomotion in the scene and interacting with its virtual objects using hand tracking and gesture recognition requires a high user effort [7, 11, 13, 15, 25]. Moreover, freehand locomotion techniques with continuous controlling constrain the user to keep his hands in the same position for the entire movement time [3, 30]. This limitation is also highlighted in [3], which proposed an empirical evaluation study of three freehand-steering locomotion techniques compared with each other and with a controller-based approach. Indeed, this study aimed to evaluate the effectiveness, efficiency, and user preference in continuously controlling the locomotion direction of palm-steering, index-steering, gaze-steering, and controller-steering techniques. In [30], authors proposed a locomotion technique based on double-hand gestures, in which the left palm was used to control

the forward and backward movement, and the right thumb was used to perform the left or right turning. The users reported a low level of perceived fatigues, reasonable satisfaction, and learning and operating efficiently. An interesting approach was proposed in [26] that focused on AR intangible digital map navigation with hand gestures. The authors conducted a well-structured study in which they explored the effect of handedness and input mapping by designing two-hybrid techniques to transit between positions smoothly. From this study, the authors reported that input-mapping transitions could reduce arms fatigue, increasing performance. A novel 3D interaction concept in VR was proposed in [21], which integrated eye gaze to select virtual objects and indirect freehand gestures to manipulate them. This idea is based on bringing direct manipulation gestures, such as two-handled scaling or pinch-to-select, to any virtual object target that the user looks at. For this purpose, the authors implemented an experimental UI system consisting of a set of application examples. The gaze and pinch techniques were appropriately calibrated to the needs of the task. Since freehand input often could invoke different operation respect the user intention, showing an ambiguous interaction effect. This problem was faced in [6], who proposed an experimental analysis to evaluate a set of techniques to disambiguate the effect of freehand manipulations in VR. In particular, they compared three input methods (hand gaze, speech, and foot tap), putting together three-timing (before, during, and after an interaction) in which options were available to resolve the ambiguity. The arm fatigue problem was faced by authors in [13] that proposed an approach based on a combination of feasible techniques called “ProxyHand” and “StickHand” to reduce this problem. The “ProxyHand” enabled the user to interact with the 3D scene in VR by keeping his arm in a comfortable position through 3D-spatial offset between his real hands and his virtual representation. “StickHand” was used where “ProxyHand” was not suitable. The gorilla-arm effect was also faced in [25], who proposed a VR low-fatigue travel technique hand-controller-based useful in a limited physical space. In this approach, the user can choose the travel orientation through a single controller, the average of both hand controllers, or the head. Furthermore, the speed can be controlled by changing the frontal plane controller angle. The gorilla-arm effect was dealt with by also estimating some useful metrics, *e.g.*, the “Consumed Endurance” proposed in [11]. In the same way, this paper [15] developed a method to estimate the maximum shoulder torque through a mid-air pointing task. However, to the best of our knowledge, there are no specific hand-and-controller-based studies where hand and controller were used to interact and locomote in 3D scenes to reduce the user arm fatigue.

3 Background

The proposed approach, which consists of the 3D scene and the case study (see Sect. 5), was developed using the Unity 3D game engine. In addition, Unity 3D allowed us to integrate the Steam VR frameworks for the integration of the HMD, its controllers, and the Leap Motion SDK for hand tracking and gesture

recognition (see Fig. 2). Users can immerse in VR scene through the HTC Vive HMD (first version), which consists of a 3.6 inch AMOLED with 2160×1200 pixels (1080×1200 per eye), a refresh rate 90Hz, and 110 degrees as a field of view. Thanks to the *Eye Relie* technology HTC Vive allows users to adjust the interpupillary distance and the distance of the lens to better adapt to its specifications. The locomotion of the user inside the 3D scene was performed through a single HTC Vive controller (see Fig. 2), which is based on a 6-DOF and is associated with a specific hand (left and right). HMD and the controller are tracked from the base stations defining a room scale [20] in which the user can move freely. Leap Motion is a well-known infrared-based device useful to track the user’s hands and perform gesture recognition from them. The compact size of the Leap Motion device allows it to be mounted directly on the HMD, enabling the users to visualize their hands in the virtual scene and increasing their VR immersion feeling. As explained in Sect. 5, our multimodal interaction approach was tested on the 3D Graph visualization. To generate and export the graph structure for the proposed application, we used a well-known software called Gephi [2]. In particular, through Gephi, we defined the colors, the sizes, the positions of the nodes, and the adjacent nodes that define edges. Furthermore, we used the Force Atlas 2 [14] to assign a 3D layout to the graph. The next operation consists of exporting the graph description as *.gexf*, which is the standard Gephi format based on XML. Subsequently, the *.gexf* file was imported in our 3D scene through a specially developed parsing function. We also used a random graph generation Gephi function to create large graphs to test the robustness of our 3D application with a growing variable number of edges and nodes. The 3D Graph scene was created starting from the *.gexf* file, which was scanned from a *PlacePrefabPoint* component, a particular Unity 3D component (prefab) we developed. For each node found, it creates a specific prefab to which the graphic features are associated (*e.g.*, sphere, color, position, *etc.*). Furthermore, *PlacePrefabPoint* creates an empty adjacency list for each node, which will be filled from another prefab called *PlacePrefabEdge* with the connected nodes defined in the *.gexf* file, and creating the edges.

4 Interaction System

VR immersion in our developed 3D scene is allowed through the HTC Vive HMD. The locomotion is performed through one of its controllers, which is kept in one user’s hand. Instead, the interaction with the 3D scene components is allowed through the other user’s hand tracked from the Leap Motion controller, as shown in Fig. 1. In this section, we describe the fundamental part of our multimodal approach.

4.1 Locomotion

HTC Vive 6-DOF controller allows the user locomotion through the 3D scene. As can be seen in Fig. 2, the user has to interact with the trackpad of the

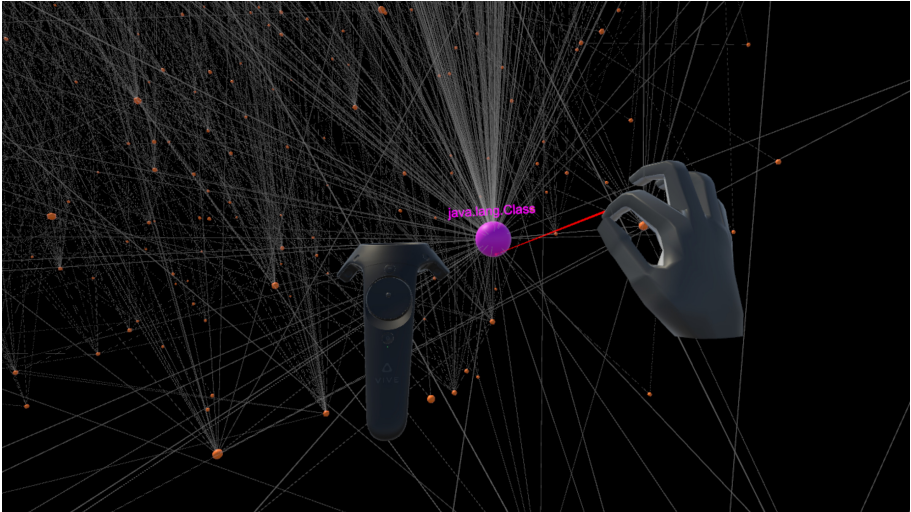


Fig. 1. The image shows the rendering of a 3D graph with its devices used to interact with the scene. The controller and the tracked hand work jointly. When a node is selected, its color changes to purple and the ray becomes red. (Color figure online)

controller to provide the locomotion direction. In particular, the user can move along the horizontal axis: forward, backward, left and right in flight mode. The vertical movement is possible through the HMD forward direction by pointing and pressing the forward trackpad button. Furthermore, the Grip button (see left part of Fig. 2) allows the user to increase the locomotion speed.

The buttons of the controller are used to activate and interact with the application menu, which allows changing the option setting (*e.g.*, max user speed movement and max graph plot scale) of the application. Furthermore, the menu allows the user to visualize and navigate the list of nodes, select one node and teleport in front of it. Another menu feature concerns the possibility to view details of the selected nodes, *i.e.*, their labels.

4.2 Interaction

The interaction was implemented through the ray casting technique [24] by setting the ray collision distance as *inf*. In particular, to select one or more desired nodes of the 3D Graph, we used a simple pinch gesture by considering the thumb and the index fingers (see Fig. 2). In particular, the node selection feature is activated whether the distance between the index and thumb fingers is between 2 cm and 2.5 cm. To select multiple nodes, the user can use the same gesture and look at the desired nodes while keeping the hand performing the pinch gesture in the HMD field of view because losing track of the hand will deactivate this function. The ray is cast from the HMD position in the same direction as the camera. For ease of use' sake, the ray is rendered using hand position instead of HMD

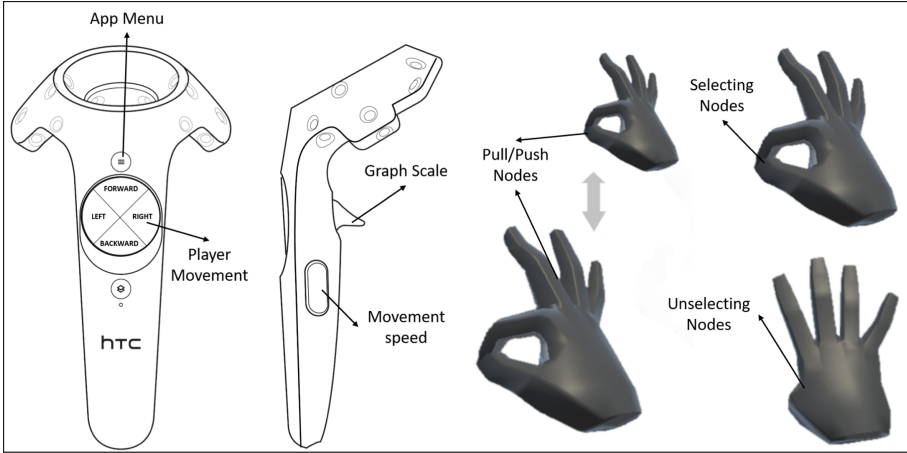


Fig. 2. The scheme of the multimodal approach. On the left, the features associated with the HTC Vive controller helpful to interact with the menu and for the locomotion in the scene. On the right, the gestures associated with hand tracking using the leap motion controller to interact with the scene components (the nodes).

position, as can be seen in Fig. 1. To ensure a good level of stability and accuracy of both the displayed and used ray, we included a queue that keeps track of the average of the last 15 positions. If the user must deselect the nodes, the tracked hand has to be fully opened with fingers toward the up direction, except the thumb finger, which has to be closed towards the palm. When the nodes are selected, their color change to purple. (see Fig. 3). For simplicity, when the ray points to an unselected node, the color of the ray changes from green to red. Selected nodes can be moved and placed in other scene positions. To move the nodes, the user has to activate the node selection feature. Furthermore, when the ray is directed toward the already selected node (purple), the ray color change to yellow and node follow the HMD position until the user releases the pinch. The last feature concern the graph scale, which can be activated from the user, holding down the trigger button of the HMD controller (see Fig. 2) and move the tracked hand and the HMD controller away from each other.

5 Case Study

The proposed navigation and interaction system finds fertile ground in virtual scenes characterized by complex data structures, such as 3D graphs. Indeed, interacting and navigating these 3D structures in VR requires a massive user effort, primarily when they represent highly interconnected data.

5.1 Graph Visualization

To explore the 3D graph, the user has to fly through the scene, and he can move in 6 directions (left, right, up, down, forward, and backward) (see Sect. 4.1).

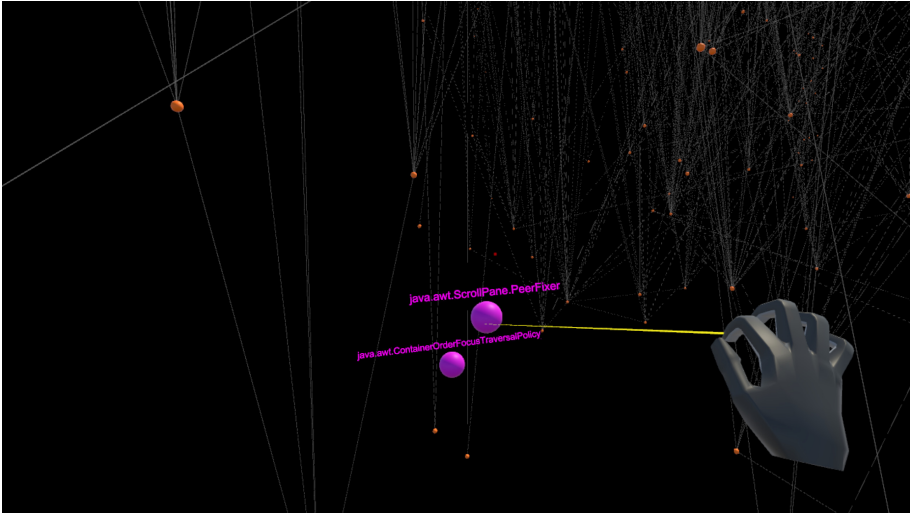


Fig. 3. When the ray hits one or more nodes already selected, the ray color becomes yellow, and the user can move the nodes. (Color figure online)

As often happens in 3D graph visualization [4], nodes are displayed by spheres and edges through rendered lines (see Fig. 4). The use of the physical object as a sphere to represent nodes is required to allow user interaction. Instead, the edges are non-physical object because not allow user interaction. In this way, we reduced the polygonal complexity of the edges, increasing performance and allowing the system to represent in 3D as many nodes as possible. The graph and its structure (node colors, sizes, positions, labels and edges) were defined in a *.gexf* XML-based format (Fig. 5), which is a standard file format of the well-known Gephi [2] graph visualization software.

When all nodes are visible from the VR camera, the worst performances are obtained in terms of rendering time. This situation occurs when the user is placed in the center of the 3D graph or looks at the whole graph from the outside of it. To solve this problem, we applied a common computer graphics approach called Level Of Details (LOD) (see the blue block in Fig. 5). This approach adapts the geometric complexity of the 3D objects automatically based on their distance from the camera. In our application, we defined 3 LODs and an occlusion culling level. This last level allows preventing the rendering for all nodes that are completely hidden from view. In this way, we further improved the rendering time performance.

With the LOD 0, the sphere mesh is high poly and very heavy. A label is associated with each node as a child of the sphere. To allow the user to visualize such labels, we rotate the node hierarchy (sphere and label) toward the VR camera. At LOD 0, the rotation is computationally expensive, and for this reason, we allow the rotation every 4 seconds to reduce the GPU workload. In the LOD 1, the node hierarchy is not present because we assumed the label

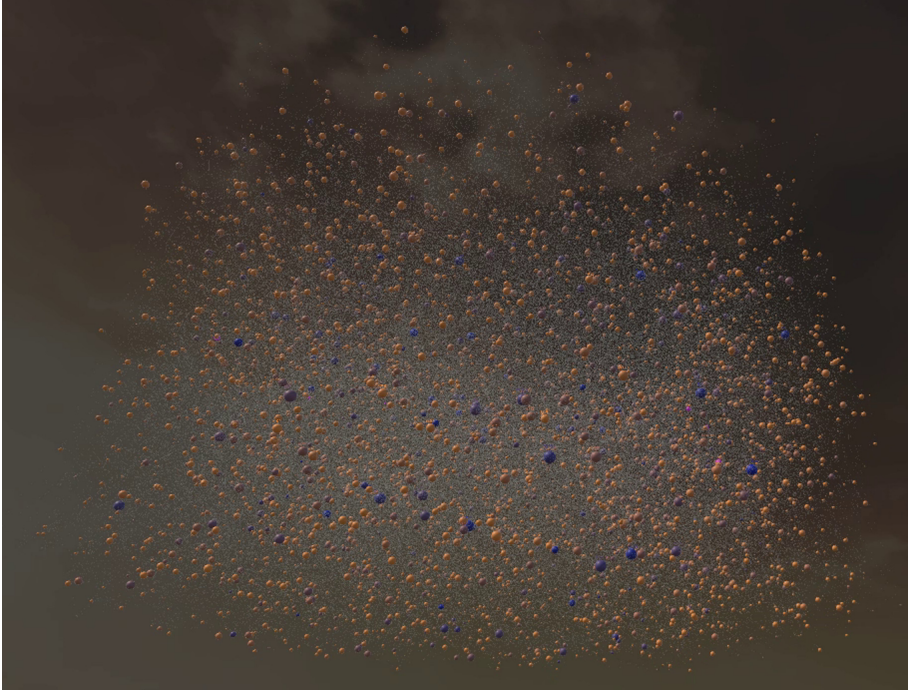


Fig. 4. Graph 3D view: nodes are represented from spheres (colors, sizes, positions, and edges are defined in the *.gexf* file); edges are represented from rendered lines. (Color figure online)

was not visible beyond a certain distance from the user. For this reason, we consider only a simplified sphere with fewer polygons and to retrieve further computing resources, for this LOD 1 and the LOD 2, we removed the rotation of the sphere toward the user. In particular, in the LOD 2, we foresee using a low poly sphere because the distance between the user and the nodes makes this simplification unnoticeable. Finally, the last is a culling level, in which the nodes are completely hidden, decreasing the GPU workload. LOD activation distance was identified based on the percentage of the camera portion occupied from each node. Based on empirical evidence, we established that the LOD 0 was activated when the percentage is between 100% and 15%, the LOD 1 was activated when the percentage is between 15% and 3%. Finally, the LOD 2 was triggered when the percentage is between 3% and 1%, and the culling level is activated when the percentage is less than 1%. Furthermore, we increased our application performance by considering the edges' behaviour during the nodes' movement. Indeed, when a node is selected and moved by the user, the edges are not rendered but are hidden until the selected node is released in its final position. Only at the end of the movement will the edges' length be recomputed, and next, they will be displayed again. This decrease the recomputing positions

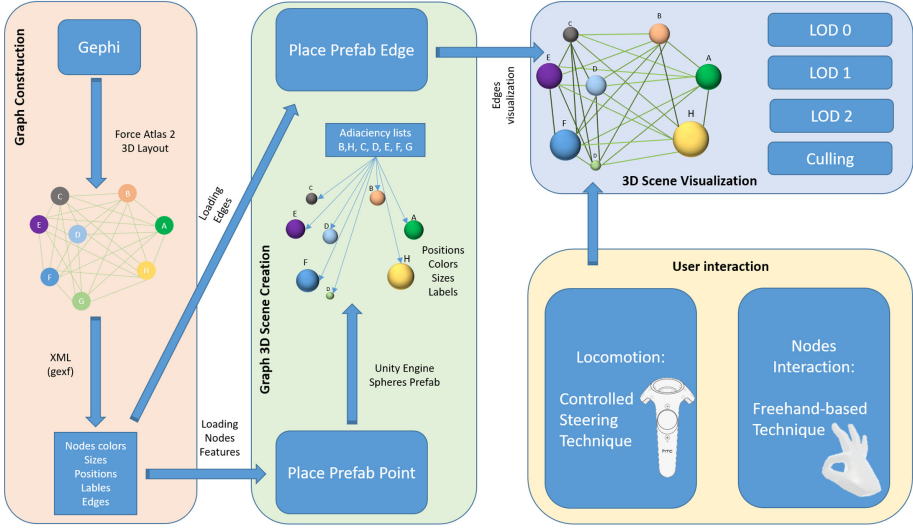


Fig. 5. The workflow design. Orange block shows the graph features definition using Gephi and storing it in an XML document. The green block shows the 3D graph building using Unity 3D and two developed prefabs. The blue block shows the 3D scene optimization, and finally, the yellow block shows the user HCI and locomotion techniques. (Color figure online)

effort for the edges in real-time by carrying out this operation only when the node movement was done. We reported two experiments with two different graph configurations in terms of nodes and edges in Table 1.

Table 1. These tests are performed using a workstation equipped with an Intel 8700k CPU, a RAM with 16 GB memory, and one NVidia GTX 1080 Ti GPU.

Nodes	Edges	VR FPS	Desktop FPS
1800	8000	50	71
6000	60000	45	57

5.2 Application Features

When starting the application, the user selects the *.gexf* that contains the information of the graph structure, as discussed in Sect. 5.1. The user can move freely in the 3D graph using the flight mode, increasing or decreasing the movement speed, and without direction and distance constraint. To select the desired nodes, the user can use a viewfinder and performs the pinch gesture to activate the ray casting, as mentioned in Sect. 4.2. By continuously keeping the pinch

gesture active, the user can select several nodes using the viewfinder to direct the selection action. The viewfinder is also activated when the user performing the deselection action by pointing the HMD direction toward the desired node (previously selected). To deselect more than one node, the user must perform the deselection gesture pointing towards an area of the scene without nodes. Figure 6 shows the application menu, which allows the user to choose from several features. One of them consists in the list of selected nodes visualization through the *Go To Node* button. The system allows the user to teleport his position in front of the selected node taken from the menu nodes list. When the application menu is active, four buttons are rendered on the VR visualized controller. The button *Cluster* allows the user to gather the selected nodes in only one node with a label containing a progressive number as text. The user can merge among them also clusters of clusters or clusters and nodes together. *Uncluster* button allows the user to split the nodes of a selected cluster. If the user has selected several clusters, the *Uncluster* button enables the user to separate them. Finally, the button *Menu* allows the user to close the menu, and the button *Details* enable the user to view the selected node label or the list of nodes contained in the cluster.



Fig. 6. The application menu. On the controller mesh the *Cluster*, *Uncluster*, *Details*, and *Menu* buttons appeared. The user can interact with them to perform the clustering, unclustering, view node or cluster details, and close the menu. The image on the right shows how to consult the list of selected nodes. The user can select a node from the list and teleport in front of the desired node through the *Go To Node* button.

6 Conclusions and Future Works

We presented a multimodal approach in VR to allow the users to interact with the virtual scene using hands gestured tracked by the Leap Motion controller and locomote through the virtual 3D scene using a 6-DOF controller. As a case study, we implemented a VR 3D Graph Visualization application developed with Unity 3D game engine and provided a description of hand-based methods to interact with the nodes, visualize their details, and locomote in the 3D Graph scene using the controller. Furthermore, we proposed an optimization strategy to render as many as possible nodes and edges by preserving the computational performances. To the best of our knowledge, the proposed work is the first to evaluate the combination of natural interaction through hand gesture recognition and controller-based navigation mode to decrease user arm fatigue. The next step consists of experimentation through comparison among several HCI techniques. In particular, we are working on an extension of the proposed application, which includes combining different types of devices to interact and locomote with and in a virtual scene.

In addition, the hand tracking could be improved using an end-to-end deep learning-based system, which replaces the Leap Motion controller [9]. For example, a deep neural network for RGB camera image analysis could be used to overcome the infrared devices' well-known lighting problems.

References

1. Virtual Reality Applications: Guidelines to Design Natural User Interface, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 1B: 38th Computers and Information in Engineering Conference (2018). https://doi.org/10.1115/DETC2018-85867_v01BT02A029
2. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks (2009). <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154>
3. Caggianese, G., Capece, N., Erra, U., Gallo, L., Rinaldi, M.: Freehand-steering locomotion techniques for immersive virtual environments: a comparative evaluation. *Int. J. Hum. Comput. Interact.* **36**(18), 1734–1755 (2020). <https://doi.org/10.1080/10447318.2020.1785151>
4. Capece, N., Erra, U., Grippa, J.: GraphVR: a virtual reality tool for the exploration of graphs with HTC Vive system. In: 2018 22nd International Conference Information Visualisation (IV), pp. 448–453 (2018)
5. Capece, N., Erra, U., Gruosso, M., Anastasio, M.: Archaeo puzzle: an educational game using natural user interface for historical artifacts. In: Spagnuolo, M., Melero, F.J. (eds.) Eurographics Workshop on Graphics and Cultural Heritage. The Eurographics Association (2020)
6. Chen, D.L., Balakrishnan, R., Grossman, T.: Disambiguation techniques for free-hand object manipulations in virtual reality. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 285–292 (2020)

7. De Chiara, R., Di Santo, V., Erra, U., Scarano, V.: Real positioning in virtual environments using game engines, vol. 1, pp. 203–208 (2007), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84878192833&partnerID=40&md5=83a6d6543f0358d6bd23a9882c42a65f>, cited By 12
8. Falcao, C., Lemos, A.C., Soares, M.: Evaluation of natural user interface: a usability study based on the leap motion device. *Procedia Manuf.* **3**, 5490–5495 (2015), <https://www.sciencedirect.com/science/article/pii/S2351978915006988>, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015
9. Grusso, M., Capece, N., Erra, U., Angiolillo, F.: A preliminary investigation into a deep learning implementation for hand tracking on mobile devices. In: 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), pp. 380–385 (2020)
10. Hillmann, C.: Comparing the Gear VR, Oculus Go, and Oculus Quest, pp. 141–167. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-4360-2_5
11. Hincapié-Ramos, J.D., Guo, X., Moghadasian, P., Irani, P.: Consumed endurance: a metric to quantify arm fatigue of mid-air interactions, pp. 1063–1072. CHI 2014. Association for Computing Machinery, New York, NY, USA (2014). <https://doi.org/10.1145/2556288.2557130>
12. Hooks, K., Ferguson, W., Morillo, P., Cruz-Neira, C.: Evaluating the user experience of omnidirectional VR walking simulators. *Entertainment Comput.* **34**, 100352 (2020). <https://www.sciencedirect.com/science/article/pii/S1875952119301284>
13. Iqbal, H., Latif, S., Yan, Y., Yu, C., Shi, Y.: Reducing arm fatigue in virtual reality by introducing 3D-spatial offset. *IEEE Access* **9**, 64085–64104 (2021)
14. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLOS ONE* **9**(6), 1–12 (2014). <https://doi.org/10.1371/journal.pone.0098679>
15. Jang, S., Stuerzlinger, W., Ambike, S., Ramani, K.: Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3328–3339. CHI 2017. Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3025453.3025523>
16. Lee, P.W., Wang, H.Y., Tung, Y.C., Lin, J.W., Valstar, A.: Transaction: hand-based interaction for playing a game within a virtual reality game. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, pp. 73–76. CHI EA 2015. Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2702613.2728655>
17. Li, Y., Huang, J., Tian, F., Wang, H.A., Dai, G.Z.: Gesture interaction in virtual reality. *Virtual Reality Intell. Hardware* **1**(1), 84–112 (2019). <https://www.sciencedirect.com/science/article/pii/S2096579619300075>
18. McMahan, R.P., Lai, C., Pal, S.K.: Interaction fidelity: the uncanny valley of virtual reality interactions. In: Lackey, S., Shumaker, R. (eds.) VAMR 2016. LNCS, vol. 9740, pp. 59–70. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39907-2_6
19. Metsis, V., Smith, K.S., Gobert, D.: Integration of virtual reality with an omnidirectional treadmill system for multi-directional balance skills intervention. In: 2017 International Symposium on Wearable Robotics and Rehabilitation (WeRob), pp. 1–2 (2017)

20. Peer, A., Ponto, K.: Evaluating perceived distance measures in room-scale spaces using consumer-grade head mounted displays. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI), pp. 83–86, March 2017
21. Pfeuffer, K., Mayer, B., Mardanbegi, D., Gellersen, H.: Gaze + pinch interaction in virtual reality. In: Proceedings of the 5th Symposium on Spatial User Interaction, pp. 99–108. SUI 2017. Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3131277.3132180>
22. Piumsomboon, T., Lee, G., Lindeman, R.W., Billinghamurst, M.: Exploring natural eye-gaze-based interaction for immersive virtual reality. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI), pp. 36–39 (2017)
23. Pyo, S.H., Lee, H.S., Phu, B.M., Park, S.J., Yoon, J.W.: Development of an fast-omnidirectional treadmill (F-ODT) for immersive locomotion interface. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 760–766 (2018)
24. Roth, S.D.: Ray casting for modeling solids. *Comput. Graph. Image Process.* **18**(2), 109–144 (1982). <https://www.sciencedirect.com/science/article/pii/0146664X82901691>
25. Sarupuri, B., Chipana, M.L., Lindeman, R.W.: Trigger walking: A low-fatigue travel technique for immersive virtual reality. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI). pp. 227–228 (2017)
26. Satriadi, K.A., Ens, B., Cordeil, M., Jenny, B., Czauderna, T., Willett, W.: Augmented reality map navigation with freehand gestures. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 593–603 (2019)
27. Szabó, B.K.: Interaction in an immersive virtual reality application. In: 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 35–40 (2019)
28. Voigt-Antons, J.N., Kojic, T., Ali, D., Möller, S.: Influence of hand tracking as a way of interaction in virtual reality on user experience. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–4 (2020)
29. Wiedemann, D.P., Passmore, P.J., Moar, M.: An experiment design: investigating VR locomotion & virtual object interaction mechanics (2017)
30. Zhang, F., Chu, S., Pan, R., Ji, N., Xi, L.: Double hand-gesture interaction for walk-through in VR environment. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), pp. 539–544 (2017)