Graduação em Ciência da Computação

A COMPARATIVE EVALUATION OF DIRECT HAND AND WAND INTERACTIONS ON CONSUMER DEVICES

By

EDUARDO HENRIQUE DE MESQUITA RODRIGUES

B.Sc. Dissertation

Federal University of Pernambuco
graduacao@cin.ufpe.br
www.cin.ufpe.br/~graduacao

RECIFE/2018

Eduardo Henrique de Mesquita Rodrigues

A COMPARATIVE EVALUATION OF DIRECT HAND AND WAND INTERACTIONS ON CONSUMER DEVICES

A B.Sc. Dissertation presented to the Center for Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Bachelor in Computer Science.

Advisor: Veronica Teichrieb
Co-Advisor: Lucas Silva Figueiredo

RECIFE
2018

Monografia apresentada por **Eduardo Henrique de Mesquita Rodrigues** ao programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **A comparative evaluation of direct hand and wand interactions on consumer devices**, orientada pela **Prof. Veronica Teichrieb** e aprovada pela banca examinadora formada pelo professor:

Prof. Giordano Ribeiro Eulalio Cabral
Centro de Informática/UFPE

RECIFE
2018

Lizi.

Acknowledgements

I would like to say thank you to all excellent professors from the Center of Informatics, for being essential in my formation, to my beloved girlfriend, to my friends, my family, and my dog. Oh, and to my little boss too! (͡° ͜ʖ ͡°)

You lock the door and throw away the key, there's someone in my head but it's not me.

—PINK FLOYD

Resumo

Com a crescente popularização de Headsets de Realidade Virtual, surge uma demanda sobre como melhor utilizar esses dispositivos nessas aplicações. Este trabalho propõe comparar o uso de dois dispositivos diferentes de entrada para ambientes virtuais; joysticks e mãos. Nossa equipe conduziu experimentos utilizando dispositivos que estão disponíveis atualmente no mercado para uso por consumidores finais, sendo estes o Leap Motion e o HTC Vive, buscando entender qual desses dispositivos de entrada mais populares se adequam as necessidades dos usuários. Cinco cenários distintos foram testados, explorando interações próximas e à distância. A avaliação foi dividida em 3 passos: uma avaliação de perfil dos usuários, uma avaliação de performance de execução das tarefas, e um questionário para medir a escala de usabilidade do sistema. Os resultados obtidos mostraram que, mesmo apresentando uma menor precisão na execução das tarefas, a interface natural de interação propiciada pelo uso de uma representação virtual das mãos do usuário, ganhou a preferência quando interagindo com elementos virtuais mais próximos. Para interações a distância essa técnica ainda precisa de melhorias.

Palavras-chave: Interação com mãos e joysticks, Ray casting, Avaliação de dispositivos de entrada

Abstract

Along with the popularization of VR Head Mounted Displays, there is an increasing demand for understanding how to use these devices within those applications. This work evaluates the use of two input techniques for VR applications: wands and hands. We perform experiments using consumer devices (Leap Motion Controller and HTC Vive), aiming at understanding how popular hardware respond to users' needs. Five distinct scenarios were tested, exploring both near and far object interaction. The evaluation was divided into three steps: user profile evaluation, system performance evaluation, and System Usability Scale questionnaire. The results showed that even with a lower task accuracy, the natural interaction interface, provided by using a hand representation on the virtual world, gained user's preference when interacting with virtual elements that were close to user. For distant object interaction, it still needs some improvements.

Keywords: Hand and Wand interaction, Ray casting, Input devices evaluation

List of Figures

Contents

1

Introduction

While Designers and developers face a large set of possibilities regarding UI decisions for both visual and input strategies when creating Virtual Reality (VR) applications. Different input devices can be considered, including hand tracking technologies and techniques, motion-based joysticks/wands and gaze trackers. Depending on the distance between interaction target and user, different methods can be used as well. Nearby objects can be reached by hand touch or wand contact, while farther elements can be selected using rays casted from a joystick or user's gaze.

1.1 Consumer Devices as Input Options

IDC estimates over 2 million VR/AR headsets were shipped on 2017[1], being more than 98% of those VR headsets. Along with the popularization of VR Head Mounted Displays (HMDs), wands are presented as input options for all the main sellers. Examples are the PS Move for PlayStation VR, the Vive Controllers for HTC Vive and the Oculus Touch for Oculus Rift. In particular, Oculus Touch also tracks the state of some fingers and displays a corresponding virtual hand for the user in VR. Hand tracking alternatives are also present in VR industry. Leap, for example, offers a bare hands tracking device (its Leap Motion controller) that can be placed in front of an HMD and provide a full hand tracking solution for VR applications. Leap also provides software integration with Oculus Rift and HTC Vive.

Considering these are available commercial solutions, there is an increasing demand for understanding how to use these devices within VR applications. Besides, while there are several tasks in VR applications that can be accomplished by using both input methods (hands and wands), there is also a lack of guidelines regarding which task each input option performs better.

In addition to the choice of input devices, there is also the choice of which virtual elements should be used for interaction. Inheriting input devices from antecedent paradigms may not apply to VR, meaning keyboards, mice, and touch screens are rarely useful for VR applications. On the other hand, previous UI elements such as buttons, sliders, windows, menus,

[1] https://www.idc.com/getdoc.jsp?containerId=prUS43021317

icons, and pointers can be inherited and easily used on VR.

Figure 1.1 shows a Mixed Reality (MR) scenario (top) by Microsoft[2] and a VR one (bottom) by Masterpiece VR[3] demonstrating the use of virtual elements representing UI elements: Icons are adapted to three dimensions, represented as 3D models on the scene; Pointers are also adapted, showing a ray casted from the input wand to the final position of the cursor; The remaining elements are represented similarly to their 2D versions on desktops.

Although these are not native elements of 3D worlds, the inheritance of the WIMP HINCKLEY; WIGDOR (2012) paradigm to MR environments brings some advantages. Given that users are familiarized to these components, and the industry is refining them from a long time (as discussed by Thomas Pederson PEDERSON (2006)), these components are likely to be well recognized and efficient interaction options.

Figure 1.1: Inherited UI elements placed on MR and VR applications. Top: Microsoft Hololens MR Windows environment. Bottom: Fish Sculpture by VR artist Mary Ellis using Masterpiece VR with HTC Vive.

1.2 User Space and Viewing Experience

Besides input devices and types of UI elements, the design of VR (and some MR) applications should consider additional topics. Given that the interface is not placed on a screen (like Desktops and Mobile devices) but in a 3D environment, the placement and sizing of the VR components comprehend a new set of choices. For example, the design of the interface may consider favoring the proximity of the objects to allow direct interactions, but may also take into account the Midas Touch problem JACOB (1993) and avoid placing objects near the user to minimize accidental interactions. Other recommendations include not placing “objects too high in the scene, as this forces users to raise their hands up and block their view” and objects “closer

[2]https://docs.microsoft.com/en-us/windows/mixed-reality/types-of-mixed-reality-apps
[3]https://twitter.com/masterpieceVR/status/915587251815555073

than 75 cm (within reach) may cause discomfort to some users due to the disparity between monocular lens focus and binocular aim”[4].

Therefore, the choice of the distance between user viewpoint and positions of virtual elements represents an impact on the interaction method. Elements placed within the arms reach allow interactions by contact, with either wands or hands. Elements placed at further distances are no longer reachable, and other strategies must be used, such as pointing and casting rays.

This work evaluates the use of two input techniques for VR applications: wands and hands. We perform experiments using consumer devices, aiming to understand how popular hardware respond to users’ needs. The used wand is the Vive Controller, while the hand tracking technology is the Leap Motion Controller. Both input devices are used with the HTC Vive HMD to visualize the VR content.

1.3 Set of Interactions

Five distinct interactions were tested, also exploring two distance scenarios: within reach and far away (elements placed approximately 6.5 meters away). The explored elements are listed below, contextualized with application scenarios:

1. Buttons: numbers on a phone dial interface;
2. 1D Sliders: a 1 to 100 volume picker;
3. 2D Sliders: a 16 grid color picker;
4. Icons: a 16 grid distant selector of virtual elements as icons on a background;
5. Manipulators: a grasp and release activity to organize colored cubes as children toys.

By combining these five elements it is possible to deliver a versatile interaction framework to VR applications, allowing a large range of applications from arranging icons in space to designing 3D sculptures (and choosing tools and colors as shown on Figure 1.1) and watching videos in VR and jumping to a specific point of the video length. For all cases, we assess performance, fatigue and overall user experience on both input methods and provide insightful data for creators of VR applications.

1.4 VR as Platform for Experimentation of MR Interfaces

The performed experiments consider a coupled setup between real and virtual content. Applications that present this type of coupling behavior as follows:

[4]https://medium.com/@LeapMotion/vr-design-best-practices-bb889c2dc70

- Scale is consistent between virtual elements and the real world. Usually, virtual objects are described in a measurement system used in the real world, and the user can describe elements size with reference to real-world terms.
- Orientation of virtual world and elements follows user viewpoint rotations.
- Translations performed by the user on the real world are mimicked on the virtual world, allowing to transit on real and virtual worlds at the same time.

For instance, applications based on 360° panoramas and videos may not present the translation consistency, allowing the user to look at different directions but not to move towards an element of the panorama. These cases do not share the needed properties, for example, to reach content by moving the user viewpoint. A VR game that allows the user to see small avatars in his front and control them through joystick commands presents a level of indirectness of content placement and interaction that may interfere with the user's egocentric perception of the virtual world. In these cases, direct interactions such as hand touches may not apply, and our findings cannot be extrapolated to them.

On the counterpart, this subset (coupled virtual and real elements) comprises a range of applications that share interaction inputs and metaphors. Delivering a set of interactions for coupled real and virtual environments may imply that other Augmented Reality (AR) or Mixed Reality (MR) applications can reuse the same interactions. For example, Figure 1.2 demonstrates the portability of UI elements between AR and VR applications.

1.5 Highlights

The highlights of this work are listed as follows:

- Assess user experience and performance on five different VR interfaces using two consumer input devices. A similar comparative analysis could not be found on literature.
- Propose and evaluate a ray selection technique to tackle the selection of distant VR elements using hand gestures.
- Conduct an analysis relating user profile to performance results.
- Discuss fatigue implications of interface design for the tested cases.

1.6 Structure

This paper is structured as follows. Section 2 lists the main related works. Section 3 describes the developed interaction techniques and their corresponding test scenarios. Section 4 details the obtained results and Section 5 the main analysis and discussions. Finally, Section 6 draws our conclusions and future work directions.

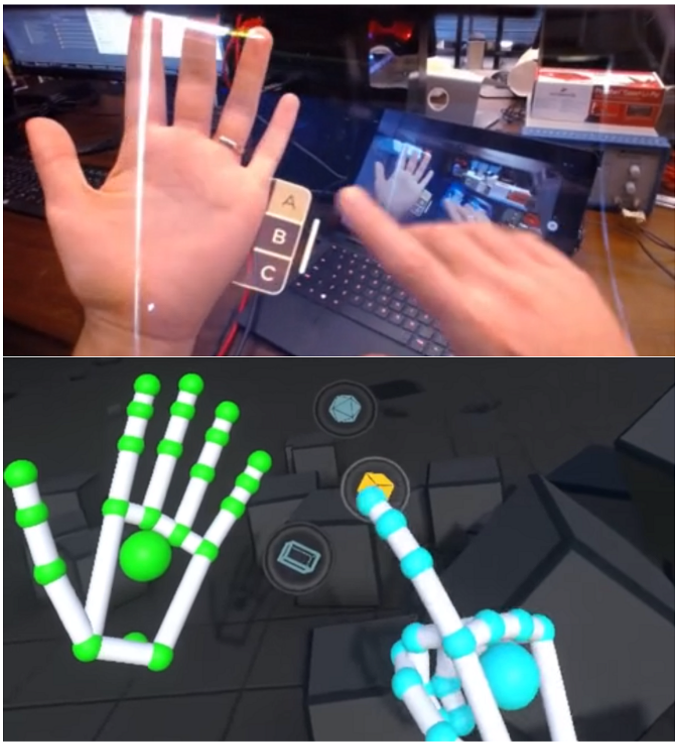


Figure 1.2: Side-palm menus shown on AR and VR applications. Top: AR productivity (office-like) utilities by Leap Motion's Project North Star, showing an extensible menu attached to the user palm with three selection options. Bottom: VR Blocks application by Leap Motion showing a palm menu with three options for object geometries.

2

Related Work

The works listed in this section are organized according to their interaction distance to the virtual objects being pointed to/manipulated. Interaction with objects far from the user and with objects in user's range are the two categories explored. The former one usually involves a larger area of interaction, such as a big screen or wall.

The need for simple pointing and clicking techniques for interacting with large displays from a distance was exposed in previous works like Vogel and Balakrishnan VOGEL; BALAKRISHNAN (2005), which evaluate five techniques for gestural pointing and clicking on objects placed far from the user. Auditory and visual feedback were provided coupled with the objects being manipulated. Its evaluations demonstrate the usability of relative hand base pointing techniques with error rates in the same low range one typically sees with status-quo devices.

Also regarding out of reach objects, Myers et al. MYERS et al. (2002) report on two studies of laser pointer interactions that answer some of the questions related to interacting with objects using rays. Myers evaluates various parameters of laser pointers, for example, the time to acquire a target and the jitter due to hand unsteadiness. It also compared 7 different ways to hold various kinds of laser pointers, 4 different ways to select objects on a large projected display, and proposed a new interaction technique that copies the area of interest from the big screen to a handheld.

Regarding the imprecision of these distant interaction methods, Nancel et al. NANCEL et al. (2015) explored the limits of existing pointing techniques and argue theoretically that they do not support high-precision pointing on ultra-walls. Some solutions to improve mid-air pointing efficiency were studied, resulting in a tunable acceleration function and a framework for dual-precision techniques. Also, a novel pointing techniques is proposed, outperforming existing techniques in controlled experiments that involve pointing difficulties never tested prior to this work. The cognitive mechanisms, strengths, and weaknesses of the proposed techniques were discussed to help interaction designers to choose the best one according to the task and equipment at hand.

Techniques using multiple rays were also studied as shown by Matulic and Vogel MATULIC; VOGEL (2018), that explored and evaluated a multi-finger raycasting design space called “multi-ray”. In their work, all five fingers can be used to generate ray intersections created by hand postures, forming 2D geometric shapes that are mapped to a set of commands making possible to perform direct manipulations that go beyond single-point interactions. A set of dynamic UI controls and operations based on the proposed technique were proposed and validated based on characteristics like projection methods, shapes, and tasks.

Also targeting large screens and objects placed far from the user, Haque et al. HAQUE; NANCEL; VOGEL (2015) describe mid-air, barehand pointing and clicking interaction technique using electromyographic (EMG) and inertial measurement unit (IMU) using a consumer armband device. The proposed technique uses enhanced pointer feedback to convey state, a custom pointer acceleration function tuned for angular inertial motion, and correction and filtering techniques to minimize side-effects when combining EMG and IMU input. The evaluation shows the technique is only 430 to 790 ms slower and has acceptable error rates for targets greater than 48 mm, demonstrating that consumer-level EMG and IMU sensing is practical for distant pointing and clicking on large displays.

The bigger the screen, the easier is for the user to move along it to select and better visualize its elements. Based on this premise, Lou et al. LOU et al. (2016) developed and evaluated a physical movements-adapted technique. This technique is based on an extension of the Pointer Acceleration approach using freehand interaction and incorporates another two adaptions based on user's physical movements. At first, it senses user's horizontal movements to gain cursor's horizontal movement in a large span. Then it obtains the distance between user and display, and generates a dynamic selection precision. The evaluation exposed significant improvements in both far object selection efficiency and accuracy.

Another technique that uses consumer-level devices is the one proposed by Carter et al. CARTER et al. (2016). This novel, distal and multi-user gestural technique focus on freehand far object and large screen interaction and is based on the principle of rhythmic path mimicry. This approach requires the user to replicate a movement of a screen-represented pattern with their hands, making it possible to interact with objects quickly and with high accuracy. They demonstrate the potential of their proposed technique as an alternative to existing ones, with key advantages for public display and multi-user applications.

Regarding distant manipulation but now focusing on tabletop devices, Banerjee et al. BANERJEE et al. (2011) present an in-air, bimanual perspective-based interaction technique called Pointable, that augments touch input on a tabletop for distant content. With the use of retroreflective markers placed on user's hand to acquire interaction information, this technique uses the user's dominant hand selects remote targets, while the nondominant hand can scale and rotate targets. Pointable allows users to manipulate out-of-reach targets, without loss of performance, while minimizing the need to lean, stand up, or involve collocated collaborators.

Also, a survey focused on distal pointing techniques regarding the possession and use of smart devices, Siddhpuria et al. SIDDHPURIA et al. (2018) compares seven distal pointing techniques one- and two-handed, using different input channels and mappings. Their results favor using a smartphone as a trackpad, but they also explore performance tradeoffs for different contexts of use.

Poupyrev et al. POUPYREV et al. (1998) evaluate immersive direct manipulation interfaces by comparing performance characteristics of interaction techniques. This paper expose their relative strengths and weaknesses, and derive design guidelines for the practical development of VE applications. A basic raycasting technique was used to evaluate distal selection task, and used a pick and place manipulation to evaluate near interaction tasks.

Fatigue is one of the biggest problems when creating an interaction interface, because it can make an accurate and fast interface impossible to use. Regarding this, Liu et al. LIU; NANCEL; VOGEL (2015) explore a relaxed arms-down position with both hands interacting with the sides of the body. They focus on interaction with large screens reachable by the user and use a Leap Motion device attached to user's leg to capture information regarding the auxiliary hand in relaxed position.

Another comparative study that is even closer to ours is the one made by Henschke et al. HENSCHKE et al. (2013). They developed a 3D pointing interface in which users are able to select, grab and drag objects in a virtual space by pointing and performing interactions with the given device, and used it to compare three different devices: 3D mouse, glove, and a wand. The system works by having the user physically pointing at the display surface with its hand and projecting a ray to a 2D point at the screen. They have found that the wand outperformed regarding time, errors and was the favorite by the users, leading them to conclude that the presence of a held object in a pointing interface changes the user's perception of the system.

Lee et al. LEE et al. (2017) use the same infrastructure from our work (Leap Motion connected to a HTC Vive) and present a technique using projective geometry to bring any near or distant window instantly to the fingertip and then to scale and position it simultaneously with a single, continuous flow of hand motion. They focus on freehand interaction, not making any use of HTC Vive's wands to evaluate the interaction.

At last, Tung et al. TUNG et al. (2015) do a more theoretical work by exploring user-defined game input for smart glasses beyond the capabilities of current sensors and focuses on the interaction in public settings. 24 participants were asked to perform common game control tasks, and the results showed that they preferred non-touch and non-handheld interaction over using handheld input devices.

3

Interaction Design and Development

This section details the two interaction techniques analyzed by this work: touching and pointing. Besides that, it provides information on how different test scenarios were chosen to better represent those two techniques.

3.1 Techniques

One of the most basic interaction tasks in VR environments is object selection. When selecting an object, the user should first be able to indicate the object of interest by either touching it with her hands or if the object is out of reach, by pointing at it. The following subsections depict how both ways of interaction were implemented using Leap Motion and HTC Vive controller.

3.1.1 Touching

Touch interaction can be defined by the intersection of some 3D point from the object with a 3D point from the user. Both can be called contact points. Regarding the 3D object to be touched, any point pertaining to its geometry can be used. In some cases, for optimizing the processing load of the application, approximations of the object's geometry are used, such as its bounding box or its simplified mesh. From user's side, the contact points to be used are selected according to the chosen controller. For instance, when using the HTC Vive wands, we have chosen the top of the physical wand as its contact point; in the case of Leap motion interaction, the position of all detected finger tips are used as multiple contact points.

In this work touch interaction was applied in two different situations: the first one comprises direct interaction with interface elements, while the second one relates to direct manipulation of objects, allowing them to be transported from a place to another by grasp/release actions.

When interacting with any interface element, such as buttons and sliders, feedback is mandatory to increase the user's sense of immersion and experience. Since in most cases haptic feedback is not available, visual or auditory cues are adopted. In this work, visual feedback was provided so that the user could easily identify which button/slider was about to be touched.

When both contact points (object's and user's) are close enough (defined by a threshold value) indicating that the user is hovering the button, it changes its color to alert the imminent interaction to happen. When touched, the button's color changes again so that it confirms the action performed by the user. An example of the three button states is shown on Figure 3.1. Used colors vary from black (far from the user) to light gray (user is effectively touching the button).

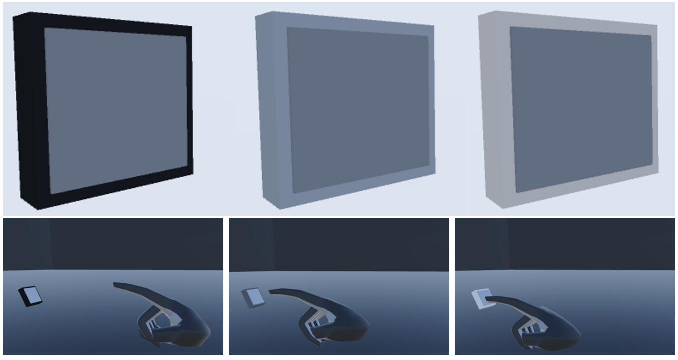


Figure 3.1: The top image shows three different button states represented by a dark to light scale, while the bottom one shows each of the three states based on the user's hand distance.

The grasp interaction using Leap Motion mimics a natural grasp action using user's hand. The user must place her hand near the object to be grasped and then simulate its grasping using any hand. After that, the object will move according to user's hand position while it is closed. To release the object in its current position, the hand must be opened and the object will automatically fall, being detached from the hand.

The grasp interaction was implemented using the HTC Vive controller through the trigger button. Once the user places the wand close enough to the object, when the trigger is pressed the object is attached to the controller. When this happens, the object's position is given as relative to the controller. When the trigger is released, the object starts falling as happens with Leap Motion interaction explained earlier.

3.1.2 Pointing

Object selection by touching it with a virtual hand and then manipulating it directly by moving the hand may be intuitive and cognitively simple, but has limited practicality BOWMAN (1998). Many virtual objects are too large to allow easy placement while close enough to touch the object. Also, it is inappropriate to force the user to move within arm's reach of an object to manipulate it, especially if the application requires multiple manipulations and efficient performance. Therefore, we are also interested in techniques that allow selection and manipulation at a distance, such as the ray casting one.

In a ray casting technique MINE (1995), a light ray emanates from the user's virtual hand. To select an object, the user intersects it with the light ray and performs a "grab" action (usually by pressing a button). She can then manipulate the object using the light ray.

For the HTC Vive Controller, the ray was implemented using the wand's center as the starting point and the top of the wand as another 3D point to establish the ray's direction.

For the Leap Motion Controller, we propose a ray casting and selection method. The ray's end point is defined by the base of user's index finger. This way it is possible to set index finger free and then use it to confirm the target selection using the pinch gesture. The ray's starting point was defined through a calibration step aiming to allow a higher angular range. The calibration procedure started by placing four cubes at the top, bottom, left and right limits of the human eye foveal region (just before the peripheral vision takes place), using 35º for the horizontal angle and 20º for the vertical angle as discussed by Poppel et al. PÖPPEL; HARVEY (1973). The goal of using these limits based on user's visual regions is to allow her to select items on the central part of the vision. Figure 3.2 shows the used eye aperture of the foveal region.

After placing the cubes at the locations described, the user comfortably moves the forearm to reach each cube, and we note the end point (base of index finger). Then we calibrate the ray's starting point (called displaced center) until locking it at a position that makes possible for the user to touch the four cubes without moving her elbow away from her body, according to Figure 3.2. This calibration is proposed based on earlier findings that show people prefer to use in-air gestures in front of their torso over gestures in front of the face (63% vs. 37%) due to concerns with social acceptance and hand/arm fatigue TUNG et al. (2015).

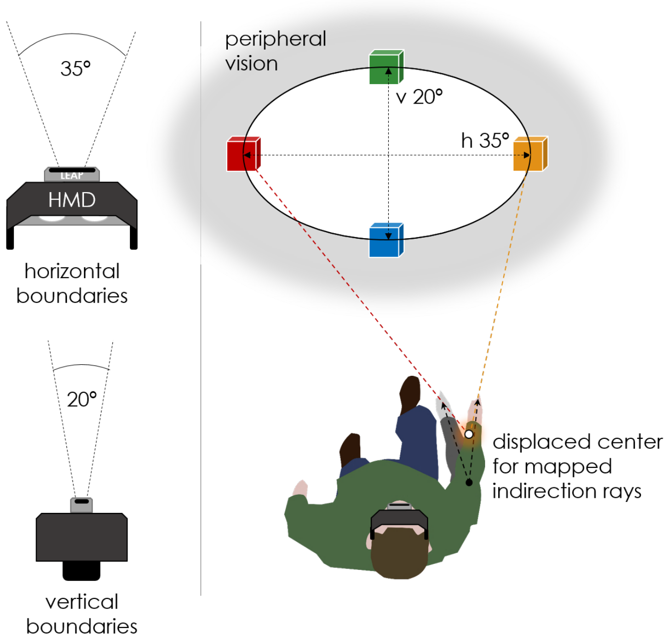


Figure 3.2: Calibration for indirect ray casting selection from user's hand.

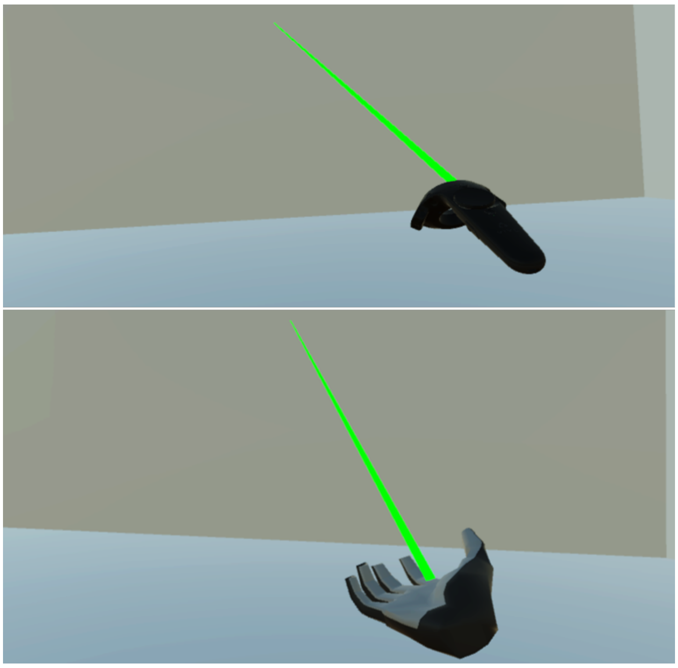


Figure 3.3: Ray casting implementations: ray casted from HTC Vive's wand (top) and ray casted from the displaced origin to the base of user's index finger using the Leap Motion Controller (bottom).

3.2 Test Scenarios

To compare the discussed techniques in both input devices, five test scenarios were created. Three of them were used to test basic interaction with 2D interface elements mapped to 3D objects, one is based on the grab interaction, and the last one is based on the ray casting technique to perform distant elements selection. Before starting the tests, users were given time to play with their virtual hands and with HTC Vive wand inside an empty room, so they could be familiarized with the virtual environment. Users were told that they should execute the touch interaction using their fingertips regarding the Leap Motion scenes and, when using the HTC Vive wands they should use the top of the ring located on top of the controller. Also, the task should be executed naturally, without hurry but also not wasting time, and they should execute it as accurate as possible. Finally, for each test scene users were able to practice three times before starting real experiments. Each of the test scenarios will be discussed as follows.

3.2.1 Button Interaction

This test scenario comprised a direct mapping of a 2D button to its 3D representation. The scene contains a 0-9 numeric keypad with two additional buttons, one for deletion and one for confirmation of the typed number. Besides that, a text panel on top of the keypad showed the numbers already typed by the user, similar to an old mobile phone interface.

The task users should perform was given as follows. At the beginning of the test, a number containing nine digits should appear above the mobile interface. The user should type the same digits in the given order. After that, the OK button should be pressed, ending the test. Every time the user pressed a button, the text on the panel was updated with the dialed number. In case the user pressed a wrong number while dialing, she should press the delete key to remove the last number dialed. To avoid visualization problems while the user's hand or wand was above the keypad, all the text presented on the scene was put on a layer above the rendered objects, so that the text overlapped every scene object and could always be seen by the user as shown on Figure 3.4.

Figure 3.4: Keypad text overlapping the scene objects.

Users were given the following instructions: At the beginning of the test two panels will appear in front of the user. The first panel contains a screen showing a number with nine digits. The second one is located below the first panel and contains an empty screen and a numeric keypad with two additional keys, "delete" and "ok", creating something similar to an old mobile phone. Every time the user pressed a numeric key on the keypad, the selected number would appear on the phone screen. If the user pressed the "delete" key, the last number on the phone screen would be deleted. The goal of this task is to type on the phone screen the same number shown in the top panel. At the end of the task the user should press the "ok" key and if the number is correct the top panel will show the message "Done!".

The complete setup for this test is shown on Figure 3.5. It is important to notice that the buttons were located in a fixed position in 3D space, while the user could move freely, positioning herself the best way she desired to make her experience more comfortable.

3.2.2 Horizontal Slider

The horizontal slider is a well known and widely used interface element in 2D interfaces, like a volume slider on a desktop operational system. To test this element in a 3D environment, we used a scenario containing a large panel with a number on it and a slider that changes this number in an interval from 0 to 100, from left to right, respectively. Above this panel, the user may see a small panel with a number interval written on it, as shown on Figure 3.6.

A number N in the interval 5-95 is randomly chosen by the application. Based on this N, the corresponding interval $((N-5)$ to $(N+5))$ appears at the top of the 2D interface. The

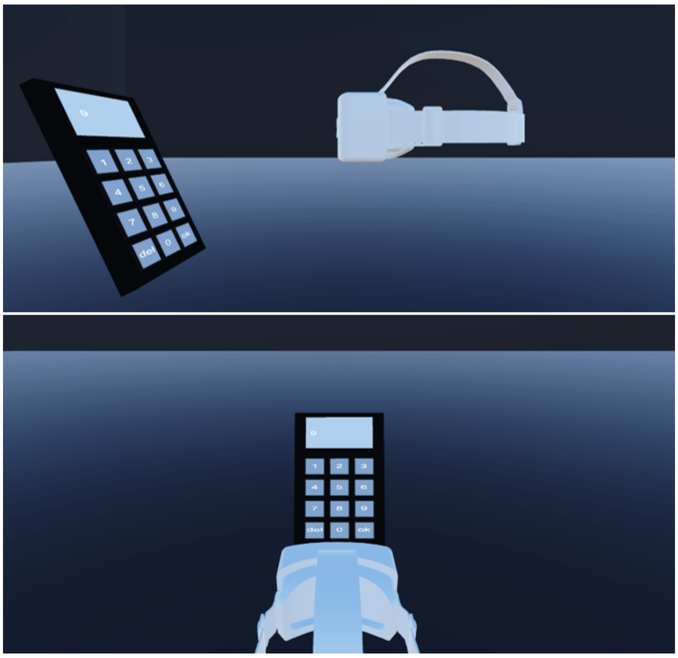


Figure 3.5: Setup for the Button interaction test. Two different views showing both old mobile phone interface and HMD.

user should move the slider so that its value is within the specified range. When the slider is released, the application verifies if the slider value is inside the target interval and repeats this five times. The task ends as soon as the user correctly places the slider inside the given intervals. It is important to notice that the slider was located in a fixed position in 3D space, while the user could move freely, positioning herself the best way she desired to make her experience more comfortable. Before the beginning of the task users were given the following instructions: Two panels would appear in front of the user. The first panel contained a number interval written on it. The second one was located bellow the first one and contained a screen with only one number on it and a horizontal slider. When the slider moved to the right, the number at the second panel screen would increase. When the slider moved to the left, the number would decrease. The user should move the slider until the number on the screen appeared within the interval shown in the first panel and then release it. Once the user released the slider on the correct position, the interval at the top panel would change its values, and the task should be repeated until the top panel showed the message “Done!”.

3.2.3 Bidimensional Slider

The bidimensional slider works as an extension of the previous test case, with the ability to move the interaction point over a square region, instead of a single horizontal line. The square is equally divided into a 4×4 matrix, with each cell showing a different color. There is a text panel above the square region showing a color name, which indicates to where the user should move the slider. The complete setup for this test is shown on Figure 3.7.

The task users should perform is given as follows. A color name was shown in the text

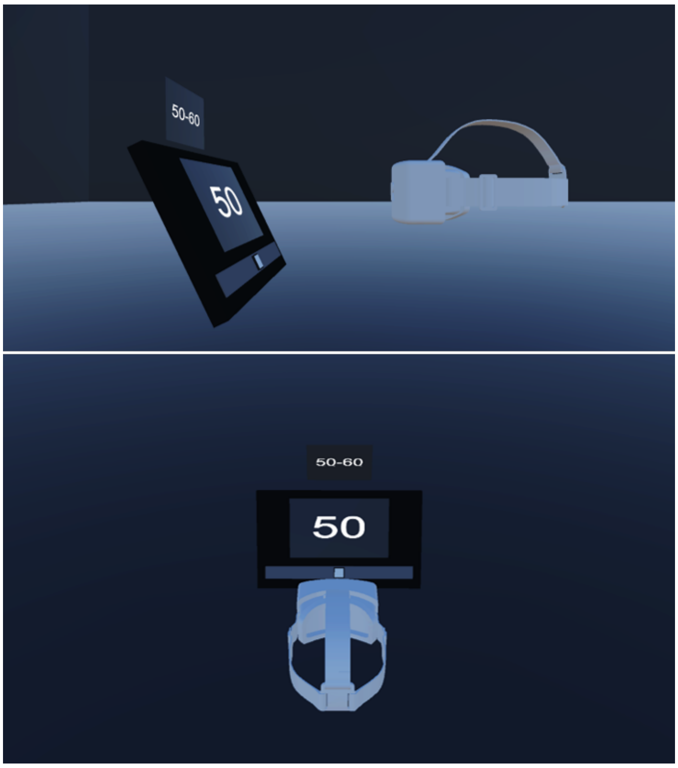


Figure 3.6: Setup for the horizontal slider interaction test. Two different views showing both 2D interface and HMD.

panel and the user must move the slider to the center of the small square containing that color. After correctly placing the slider, the task should be repeated four more times, with the target color being randomly selected each time. As soon as the user correctly placed the slider in the five indicated colors, the task ended. It is important to notice that the bidimensional slider was located in a fixed position in 3D space, while the user could move freely, positioning herself the best way she desires to make her experience more comfortable. Before the experiment starts, the following instructions were given to users: Two panels would appear in front of them. The first panel contained a color name written on it. The second panel contained a whiteboard with colored squares in a grid and a bidimensional slider that could be moved along the whiteboard. The task was to put the slider above the corresponding color exposed at the first panel. Once the slider was above the right color, the color name at the first panel would change and the task should be repeated until the first panel showed the message “Done!”.

3.2.4 Grasp and Release

This scenario was created to evaluate which interaction device was preferred for performing a task natural to human beings, such as the act of grasping and releasing objects in space. It was comprised of two tables in 3D space and eight cubes, as shown on Figure 3.8. The table to the right is white and initially contained all eight cubes over it. The second table was equally divided into four regions, each of them corresponding to a color of the cubes.

The task users should perform is given as follows. In the beginning, all cubes were

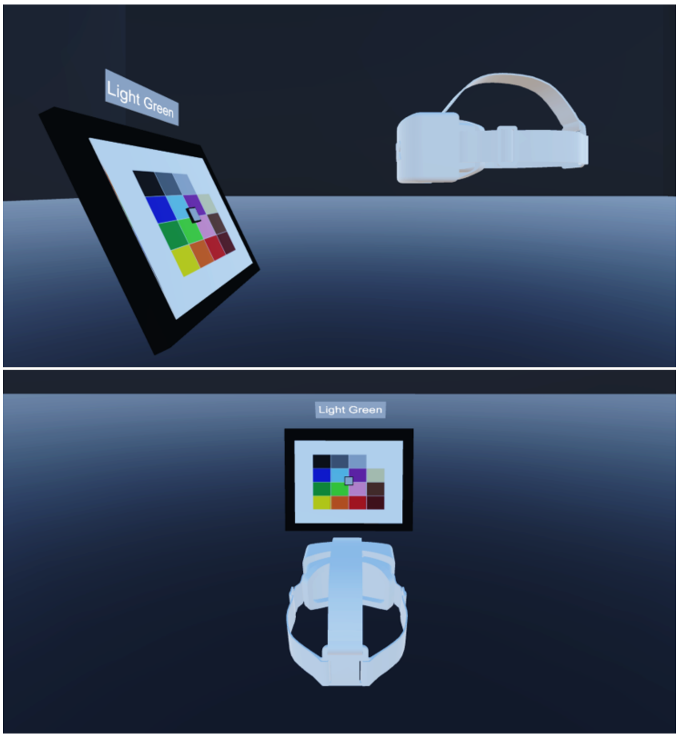


Figure 3.7: Setup for the bidimensional slider interaction test. Two different views showing both 2D interface and HMD.

randomly placed on the white table. The user should grab each one of the cubes and place them over the second table, specifically in the corresponding color. After being placed correctly the cube disappeared. In case the user released the cube over a region that did not match the cube color, the cube returned to a random position over the white table. The task ended as soon as the eight cubes were correctly placed over their corresponding colors in the second table. It is important to notice that the tables were located in a fixed position in 3D space, while the user could move freely, positioning herself the best way she desired to make the experience more comfortable.

At the beginning of the test the following instructions were given to the users: Two tables will appear in front of the user. The first table is white colored and is located more to the right. This table contains eight colored cubes on top of it. The second table is located more to the left and is divided in four equally sized and colored squares. The user should grab each one of the cubes in the first table and place it on matching color quarter of the second table. If the cube was placed on the table of a different color or falls to the ground, it will reappear on its starting position on the first table. If the cube was placed on a table of the same color the cube will disappear. The task ends when all cubes disappear.

3.2.5 Ray Casting Selection

This scenario was created specifically to test interaction with distant objects. It was comprised of a 4×4 matrix of cubes distant about 6.5 meters, all of them being out of user's

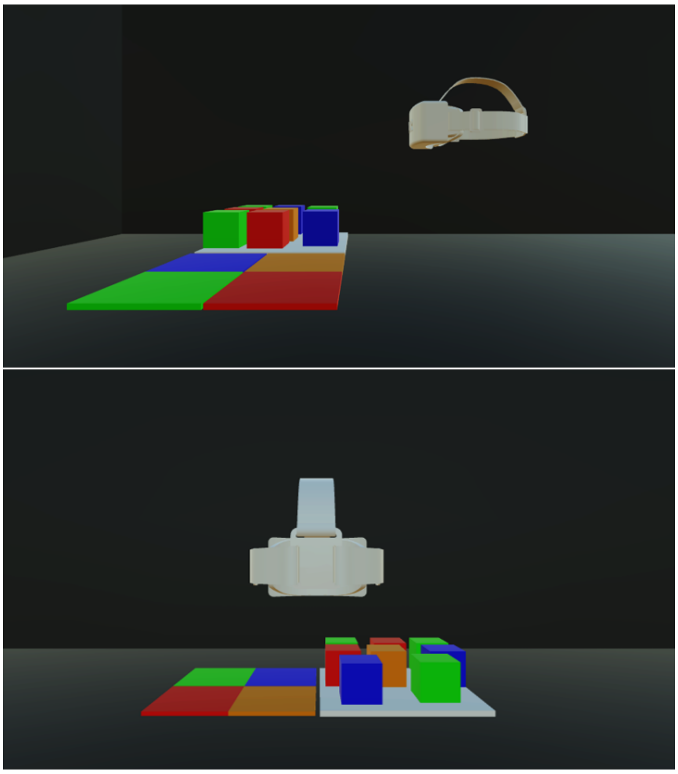


Figure 3.8: Setup for the Grasp and Release interaction test. Two different views show the two tables (colored one and white one), the 8 colored cubes and the HMD relative position.

reach, as shown on Figure 3.9.

The task users should perform is given as follows. In the beginning, one of the 16 cubes was highlighted so that the user must point to the center of such cube. To avoid the Midas Touch problem, the cube was selected with a device-dependent action (pinch action for the Leap Motion Controller and trigger button press on HTC Vive wand). After correctly selected, the cube disappeared and another one was randomly chosen. The tasks ended as soon as all cubes were "hit", according to the random order imposed by the application. There was no additional negative feedback whenever a wrong cube was hit. In fact, the user error was considered for further analysis. It is important to notice that the cubes were located in a fixed position in 3D space. Despite the fact the user could move freely, the test moderator advised the user to not do so, in a way that her position was kept approximately 6.5 meters from the cubes during the experiment.

The users were given the following instructions: Sixteen cubes would appear in a wall in front of them, organized into a 4×4 grid. One of the cubes would be highlighted. The task was to point the ray to the highlighted cube and execute the selection action. Once the right cube was selected, it would disappear and another one would be highlighted. If the wrong cube was selected nothing happened. The task ended when all cubes disappeared of the wall. Despite the fact the user could move freely, she should try to keep the starting distance of the wall by not walking in its direction.

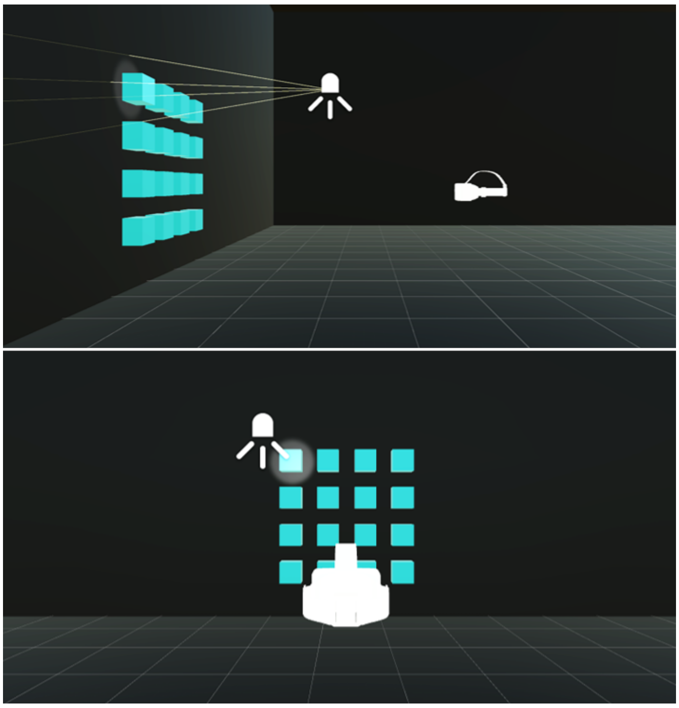


Figure 3.9: Setup for the Ray Casting Selection interaction test. Two different views show the 4×4 cube matrix (at 6.5 meters distance) and the HMD.

4

Tests and Results

The computer used to run the test cases had an Intel® Core™ i7-4790K CPU @ 4.00 GHz with 32 GB of installed RAM and a Windows 10 64-bit operating system (x64). The GPU used was a NVIDIA GeForce GTX 960 with 4 GB of RAM. All test cases were prototyped using Unity 3D version 2018.1.1f1.

The tests were performed inside a 3.78 square meter room, in which were placed the main application computer and two HTC Vive trackers. The user was placed sitted on a chair located according to the top of Figure 4.1. Chair rotation was allowed in any of the five test cases to improve users experience. Users were asked to hold the HTC Vive wands only when performing half of the tests (regarding wand interaction). For the bare hand test scenarios, they just needed to place their hands in front of them, since the Leap Motion Controller was attached to the HMD. The bottom of Figure 4.1 illustrates two different users, the left one (male) holding the HTC Vive wand, while the right one (female) was interacting using only her hands (based on Leap Motion Controller's input).

The evaluation was made with 24 participants, 14 men and 10 women, with a minimum age of 19 years, mean age of 32 years and maximum age of 63 years and was divided into three steps: 1) profile evaluation to categorize the users; 2) system performance evaluation where we measured which input device presented a faster and more accurate interaction; 3) System Usability Scale (BROOKE et al., 1996) (SUS), which comprised a Likert scale quiz that provided a global view of subjective assessments of usability. Each session took from 40 minutes to 1 hour.

The sample size for this experiment was calculated aiming 90% confidence within 4% error margin the measurement of errors regarding the percentage of wrong inputs BUKH (1992) and within 1 second for the measurement of time per input. All test scenarios are found within these constraints of confidence and error margin except for the Color Picker task (which presented an error margin of 6% for error measures) and the Volume Picker using the Leap Motion (which also presented a margin of 6% also for error measures).

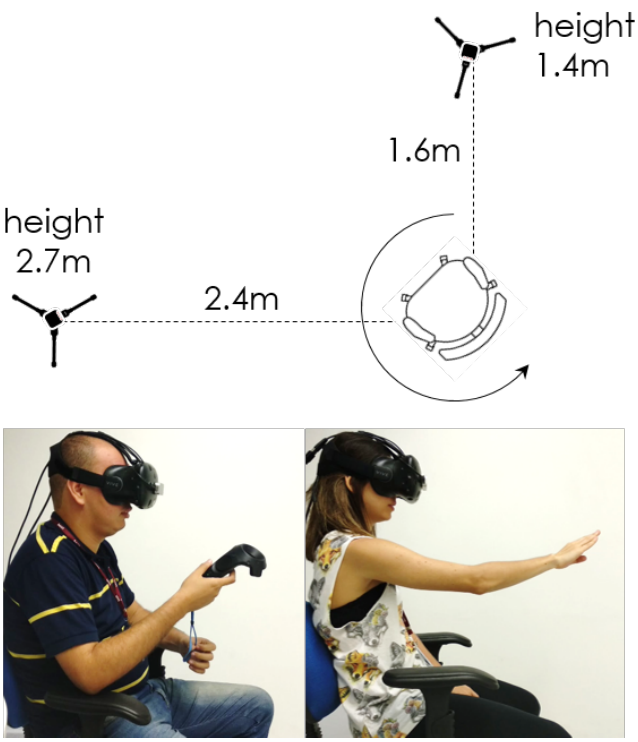


Figure 4.1: Environment setup for the tests: top view depicting the location of user position and HTC Vive trackers (top); male user interacting using the HTC Vive wand (bottom left); female user interacting using her hands (bottom right).

4.1 User Profile Evaluation

The first step, user profile evaluation, started by asking users how much time they usually spent using computers during their week, the longest time spent playing video games during a week on the last two years and how many times they used unconventional input devices to interact with 3D spaces on computers. Such questions were based on the work of Jason Jerald JERALD (2015), and the corresponding findings are exposed on Figure 4.2.

In the second step, we evaluated the interaction on the proposed scenarios by calculating the task execution time and its error rate with the goal of finding out which input device presented a better performance. The execution time was computed starting on the first user interaction with the given task, avoiding delays on the program and hardware correct initialization to affect the tests final results. The results regarding execution time and error rates for the five test scenarios are shown on Figure 4.6.

The last step was executed at the end of each task, where the users were asked to answer a SUS test. The answers were within a range of 1 to 5, varying from “strongly disagree” to “strongly agree”. Quiz results are shown on Figure 4.3.

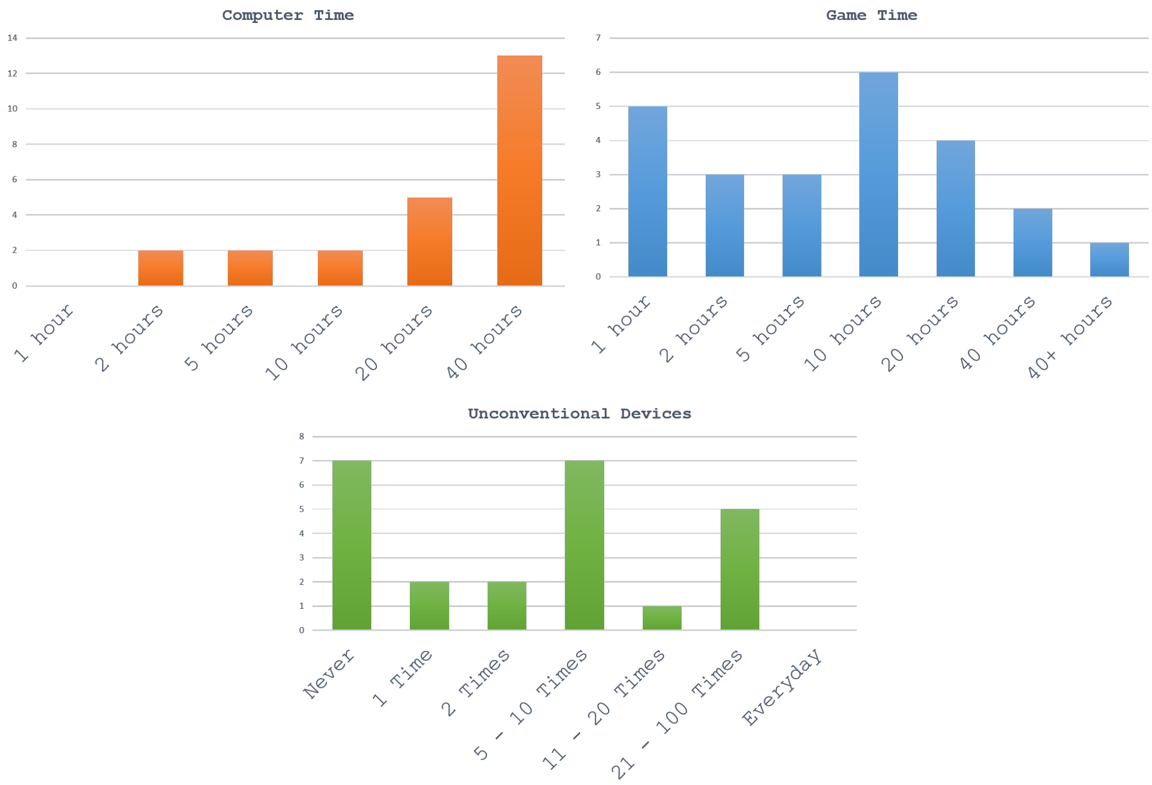


Figure 4.2: The first chart (top) shows how much time people spend using computers during a week. The second chart (middle) shows the highest time people have spent playing video games on the last 2 years. The third chart (bottom) regards the usage of unconventional input devices on computers.

4.2 SUS Application

The SUS questionnaire is a largely used tool to assess user's perception of system's usability. Similarly as the inheritance of UI elements, evaluation methods applied on applications that use earlier consolidated interaction paradigms are reused and adapted for MR applications. In addition to its application on traditional systems, SUS has also been used on a number of AR and VR studies and applications. Examples include immersiveness on VR training GRABOWSKI; JANKOWSKI (2015), VR rehabilitation solutions MELDRUM et al. (2012); LLORÉNS et al. (2015), AR educational systems LIN; CHEN; CHANG (2015), and other AR mobile applications SANTOS et al. (2014, 2015).

Although SUS questionnaire does not target specific aspects of interaction with virtual environments, it brings an overview of the user's perception regarding system's usability. Bangor and colleagues BANGOR; KORTUM; MILLER (2008) suggest slight modifications for some word choices in SUS. For instance, the authors changed the word "cumbersome" to "awkward", and "system" to "product". These changes, according to the authors, helped users to be more precise while answering the questions. In order to aid users to better relate the SUS questionnaire to their experiences with the VR interfaces, we instructed them to consider the word "system" as a representation of the currently experienced interface and interaction technique. In addition, to explicitly include the concern regarding fatigue, we added the word “tiresome” to question 8, resulting on the phrase “I found the system uncomfortable and tiresome to use”. The word “inconsistency” on question 6 (“I thought there was too much inconsistency in this system”) was also explicitly presented to users as a representation of the interaction predictability, concerning topics such as tracking failures from both Leap Motion and HTC Vive Wands.

Figure 4.3 shows the results obtained regarding the SUS evaluation applied. Most of the test scenarios were evaluated above Good in the SUS score. The only scenario that obtained a grade between OK and Good was the Ray Casting Selection using the Leap Motion Controller.

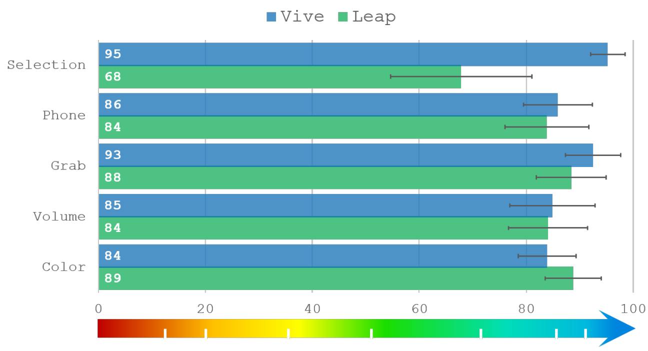


Figure 4.3: SUS scores averages and standard deviations for all test scenarios. The bottom part shows as reference adjectives placed over each corresponding score value.

4.3 Fatigue

Fatigue was assessed through observations (Figure 4.4, unstructured interviews, plus a particular question of the SUS questionnaire, which highlights the word “tiresome”. Regarding the use of specific factors and questions of the SUS questionnaire, the authors advise against making detailed scale interpretation on the basis of responses to individual test items. Nevertheless, there are additional studies that discuss this possibility. For instance, Lewis and Sauro LEWIS; SAURO (2009) suggest to treat SUS as a 2 dimensional evaluation, breaking up the 10 questions in two sets, one for *Usability* and the other for *Learnability*. In this work, we also analyze SUS item 8 separately in order to identify insights about fatigue for both input options on all five interfaces as shown on Figure 4.5.

4.4 Task Performance

For each of the test scenarios evaluated, the mean time per input was measured. This metric was calculated based on the number of actions performed by the user divided by the total time needed to complete the task. The definition of action varies according to the test scenario. For the old mobile phone test scenario, an action means the press of a button. In this case, the number of errors was calculated simply by counting how many times the “DEL” button was pressed. In the horizontal slider and bi-dimensional slider test cases, an action is considered

Figure 4.4: Captured moments during observation of user Selection (left) and Grab (right) interaction.

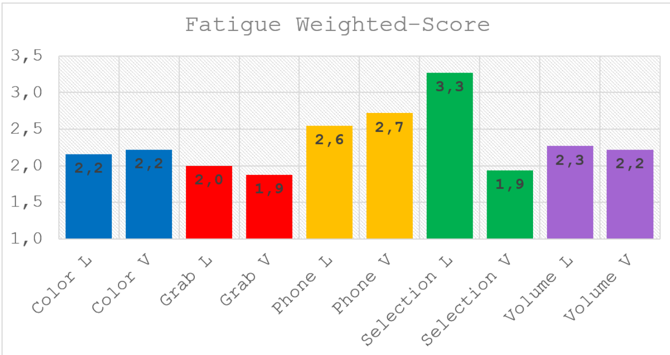
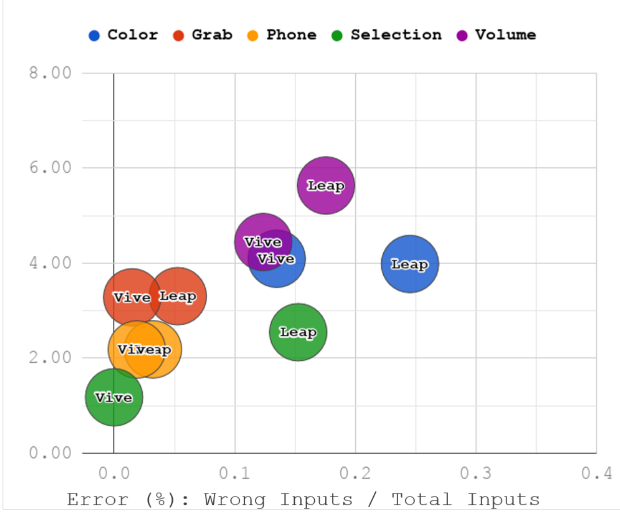


Figure 4.5: Fatigue average scores from answers to the statement "I found the system uncomfortable and tiresome to use". Scores ranged from 1 to 5, being 1 the best score (user strongly disagrees with the statement) and 5 the worst (user strongly agrees with the statement). The letters at the end of each category define that the test used either Leap Motion (L) or HTC Vive Wands (V).

as executed whenever the slider is released. In the 1D case, the error counter increases every time the user releases the slider outside the specified range. For the bi-dimensional scenario, the error counter is incremented every time the user releases the slider in a wrong color. For the Grasp and Release scenario, the action counter is incremented every time the user releases a cube. Similarly, the error counter increases every time the cube is released over a region of the table that does not match its color. For the Ray Casting Selection scenario, the action counter increases every time the user presses the trigger button (when using the wand) or performs a pinch gesture (when using the free hand interaction). The error counter increases if the ray is not intersecting a highlighted cube when the event is fired.

Figure 4.6 details the obtained results regarding the time per input metric and the amount of error perceived for each of the tested scenarios.

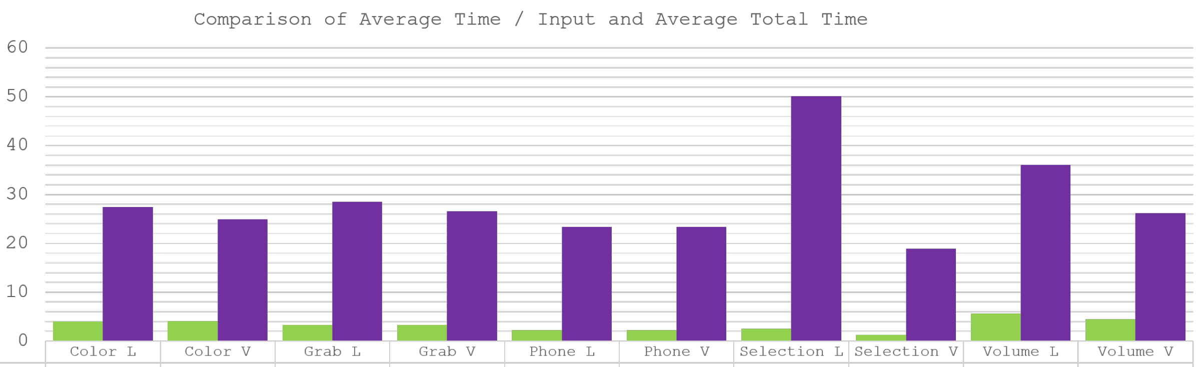
In addition to the time taken by input Figure 4.7 also shows the average of the total time taken to complete the proposed tasks. Moreover the figure shows on the bottom table the average of "Wrong Clicks" for each task and the ratio between the Total Time to complete the task and the Time Taken per Input. Since each task required an specific number of inputs the ratio between one task and another may present significant variations. The increase of wrong clicks directly affects the total time of the performed task, while it does not necessarily present a



		Error (%)		Time (sec)	
Scenario	Input	Avg.	Std. Dev.	Avg.	Std. Dev.
Color	Leap	24.53	17.52	3.98	1.2
Color	Vive	13.48	18.00	4.09	1.0
Grab	Leap	5.28	11.06	3.31	0.9
Grab	Vive	1.50	5.49	3.28	2.6
Phone	Leap	3.22	5.41	2.19	0.7
Phone	Vive	1.88	4.48	2.19	0.9
Selection	Leap	15.27	11.60	2.55	1.0
Selection	Vive	0.00	0.00	1.18	0.2
Volume	Leap	17.57	18.36	5.64	1.8
Volume	Vive	12.37	12.39	4.45	1.5

Figure 4.6: Average time per input (Y axis) *vs* average error (X axis) for each test scenario evaluated.

direct impact on the time per input. For instance, this is the case of the Selection using the Leap Motion device (average of 3.3 errors during each trial).



	Color L	Color V	Grab L	Grab V	Phone L	Phone V	Selection L	Selection V	Volume L	Volume V
Avg Time / Input	4	4.1	3.3	3.3	2.2	2.2	2.5	1.2	5.6	4.4
Avg Time (Sec)	27.4	24.9	28.5	26.6	23.4	23.4	50.1	18.9	36.1	26.2
Wrong Clicks	2	1.1	0.6	0.1	0.4	0.3	3.3	0.1	1.5	0.8
Total Time/Per Input Time	6.8	6.1	8.6	8.1	10.6	10.6	20.0	15.7	6.4	5.9

Figure 4.7: Time performance results per task and per input.

4.5 User Profile x Results

In order to understand the impact of user profile on the obtained performance results we analyzed relationships between gender, age, computer time, game time and use of unconventional devices by users. No relation was found regarding gender, computer time and the use unconventional devices for any of the performance measurements. Figure 4.8 shows the found relationship between average time of the entire task regarding the Video Game Time (top) and Age.

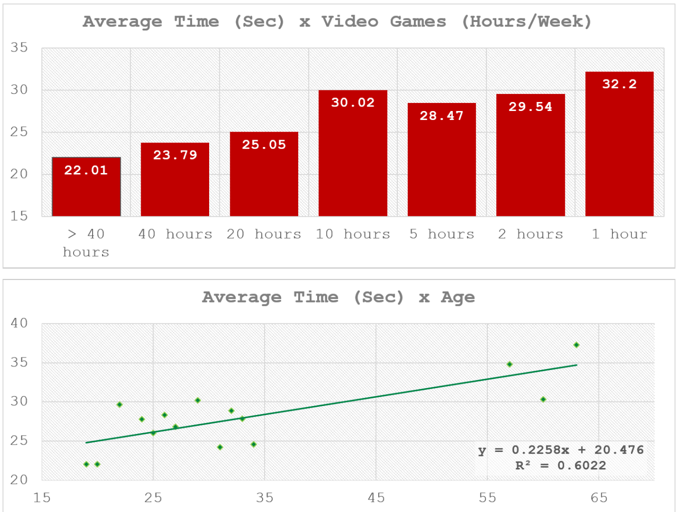


Figure 4.8: Relation between Average Time (in seconds) to complete each task and Game Time (top), and between Average Time and Age (bottom).

5

Analysis and Lessons Learned

It is worth pointing out that the performance results might be different if the size or arrangement of the elements was changed. However, this work does not consider testing these changes because this is a comparative analysis for the input devices and both devices were tested on the same conditions, meaning that the tasks should be equivalent regarding complexity.

Based on Figure 4.6, it is possible to perceive that the scenarios that presented a higher number of errors are the ones related to sliders (both 1D and 2D) interaction. This confirms a higher complexity of this action in comparison to other scenarios. Some users commented about found issues on the sliders, such as "I wanted to grasp the sliders and drag it", suggesting an engage action that maintains the link between the slider position and the users' movement even after no longer pressing the slider button. Another comment revealed a concern regarding the required pressing action of the slider and missing tactile feedback, mentioning that "given that I will have to press it, it should show some kind of resistance". Users also pointed that the Leap Motion "tracking may get in the way if you need to use the sliders with more precision or a continued use". At last, a person revealed that the use of the "volume picker was the hardest one because once I try to disengage it, it pulls off its place". Even considering that the SUS score for both input devices (Leap and Vive Controller) regarding the volume picker was higher than 80, presenting a "Good" perception of usability of this component, the activation and deactivation mechanics for the sliders may deserve a revision for future interfaces given the annoyance exposed by the users.

The color picker showed slightly faster results, and although the error with Leap Motion was higher, users demonstrated a preference towards this interface as shown by SUS results. As a user comments "although the Leap is less precise, it augments the immersion for being more real, then the execution of the task becomes more intuitive". The same user compares the Vive Controller, stating that it "is much more precise, you make it work more easily, but you need someone to instruct you about how to use it". There are other users that point on different directions, commenting, for example, that "although the hand is more natural, the controller is something common to people today and thus it does not generate a high load for learning".

The phone dial interface showed errors of 3.2% for the Leap Controller and 1.8% for the Vive Controller. SUS scores are 86 and 84 for the Vive and the Leap interface respectively, showing a near to excellent interface. Nevertheless, users pointed some difficulties such as “the cellphone button was too deep”, it required “too much effort with the hands to press the buttons” and “it is bad for people with light hands, hard to press”. In the particular case of the use of the Leap a user pointed that it was “intuitive to use but the button required me to go too deep to trigger it, and sometimes it was triggered more than once”. Two users complained about the sizing, one saying that “the keys were too near to each other and complicated to press, when they are more far apart it is easier”, and the other stated that “the cellphone was too large”. These last comments revealed that more attention should be given to the sizing of these virtual objects and this may be a personalized characteristic of the touchable UIs.

Regarding the Grab and Release activity, the results showed errors of 5.28% and 1.5%, average times of 3.31 and 3.28 seconds and SUS scores of 88 and 93 for the Leap and Vive, respectively. All users quickly understood this task goal, and this was the least commented scenario by them.

The test case that presented more errors was the Selection using the Leap Motion Controller. The proposed strategy for ray casting combined with the accuracy from the device was not good enough when applied to selection of objects placed too far from users, increasing the error rate. Another important information from Figure 4.6 is that most pair of tests (same scenario, different interaction) are close to each other. This means that in most cases users spent almost the same time interacting using both input devices. Again, the time and error difference increased when observing the Slider test scenarios and also the Selection scenario based on hand control. A comment about the use of the ray casting with the Leap was that “when you try to click it moves away”, meaning the strategy to use the base of the user’s index finger may not be enough to give freedom for a comfortable and natural pinch gesture. Another comment was that the “pinch gesture was too large, if it was smaller it would be good”. Particularly about the tracking, a user stated that “the laser from the Leap should be more firm”. While using the Vive Controller, another user pointed that “I felt like Han Solo”, and another user said “the one about shooting is funnier with the controller”, suggesting users associate that ray cast works like a shooting metaphor. It is also noticeable that the ray cast selection using the Vive Controller showed the best results for both performance and users answers on the SUS questionnaire.

Fatigue was explicitly mentioned by some users. Despite calibration efforts to allow using the selection ray while resting the arm on an armchair, most users raised the arm to direct the ray to the target and perform the pinch gesture. One user complained about resting the arm on the armchair stating that the “ray moved away from the intended position”, meanwhile we observed tracking failures from the Leap Motion. Tracking results from Leap Motion were highly associated to a specific space range in front of user’s head, which leads to fatigue and social awkwardness as pointed by Tung et al. TUNG et al. (2015), showing that if users were able to choose the region of the performed gestures it would occur in front of the chest or downwards. Ray selection can be seen on Figure 4.1, showing that while using the Vive wand a user rests his elbow on the armchair, the other user extends her arm to perform the gesture. As shown on

Figure 4.5, the worst score was obtained on the Selection task using Leap Motion, followed by the Phone task using Vive. An interesting observation is that the Grab task on both Vive and Leap presented good scores, even for users that used both hands (as illustrated in Figure 4.4) and being one of the tasks that required more motion. In part, these scores can be explained by the fact that the height where the cubes were placed was lower than the other interfaces. Possible future investigations to avoid fatigue include changing the placement of the Leap Motion device (tilting it some degrees downwards), providing additional feedback once the tracking results (of both Leap and Vive) occur out of user's line of sight, and at last adding eye gaze as an input option for pointing interest targets.

Another point to consider is to tackle the challenge of designing and developing new interfaces while dedicating specific attention to users with few or none experience with digital games. Users with an amount of over 20 hours of game time per week completed the proposed tasks in considerable less time. This may indicate effects of the natural shift regarding the production of virtual content and environments by inheriting metaphors and interfaces from the gaming community. Even the use of the Unity engine to produce the interfaces may present some impact on these results. This close relation between 3D games, game engines and VR/AR applications may produce unnoticed impacts leaving non-gamers more distant to the task of using efficiently VR interfaces. In addition, the age of the users also presented impact on the conclusion of the proposed tasks. Despite a low R^2 value, the linear regression on Figure 4.8 shows a potential relationship between user age and required time to complete each task. Given the obtained result this relationship can be considered an insight and deserves further investigation.

Another concern from the start was the shape of the wand for the contact-based interfaces (phone dial and sliders). There was no direct complaint or observation about this topic. Nevertheless, as an extrapolation of some perceived issues, probably a reviewed design of the 3D model of the joystick on the virtual scene could favor a better selection of the virtual elements. Buttons and sliders that were harder to press and release could be more easily triggered.

6

Conclusion

Based on the growth of VR/AR headsets market and considering commercial available solutions, this work focuses on evaluating different interaction devices (Vive Controllers and Leap Motion Controller) and how users respond to different interaction scenarios (1D and 2D sliders, buttons, icons selection at a distance and grasp and release of virtual objects). We also propose a ray casting technique for distant selection using the hand tracking input.

Both time per action and error metrics were evaluated for each scenario. It was possible to understand different behaviors for scenarios and devices combinations. Higher errors and time consumption were observed in both slider scenarios. Users also revealed insights about the interfaces questioning, for example, the need to press and release virtual sliders before tuning its values.

Another important finding from the test scenarios, specifically the one regarding distant object selection, is that Leap Motion Controller, due to its tracking instability, does not have enough accuracy to select distant objects by common ray casting approaches, since the errors propagate and increase with the distance to the user. This leads to an increased occurrence of false-negative and false-positive selection of objects, which decreases Leap's use potential in this kind of scenario. Nevertheless, most users reported that Leap Motion is preferred when interacting with nearby objects, since the interaction is more natural and does not depend on additional information to be used.

As future work, we intend to evaluate different alternatives of ray casting techniques focusing only on hand interaction. This will allow us to improve the results regarding the use of Leap Motion Controller for distant object interaction. We also intend to do a comparative study regarding all near and far interfaces.

References

BANERJEE, A. et al. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In: ACM INTERNATIONAL CONFERENCE ON INTERACTIVE TABLETOPS AND SURFACES. **Proceedings...** [S.l.: s.n.], 2011. p.11–20.

BANGOR, A.; KORTUM, P. T.; MILLER, J. T. An empirical evaluation of the system usability scale. **Intl. Journal of Human–Computer Interaction**, [S.l.], v.24, n.6, p.574–594, 2008.

BOWMAN, D. A. Interaction Techniques for Immersive Virtual Environments: design, evaluation, and application. **Journal of Visual Languages and Computing**, [S.l.], v.10, p.37–53, 1998.

BROOKE, J. et al. SUS-A quick and dirty usability scale. **Usability evaluation in industry**, [S.l.], v.189, n.194, p.4–7, 1996.

BUKH, P. N. D. **The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling**. [S.l.]: JSTOR, 1992.

CARTER, M. et al. Pathsync: multi-user gestural interaction with touchless rhythmic path mimicry. In: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2016. **Proceedings...** [S.l.: s.n.], 2016. p.3415–3427.

GRABOWSKI, A.; JANKOWSKI, J. Virtual Reality-based pilot training for underground coal miners. **Safety science**, [S.l.], v.72, p.310–314, 2015.

HAQUE, F.; NANCEL, M.; VOGEL, D. Myopoint: pointing and clicking using forearm mounted electromyography and inertial motion sensors. In: ANNUAL ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 33. **Proceedings...** [S.l.: s.n.], 2015. p.3653–3656.

HENSCHKE, M. et al. Wands are magic: a comparison of devices used in 3d pointing interfaces. In: IFIP CONFERENCE ON HUMAN-COMPUTER INTERACTION. **Anais...** [S.l.: s.n.], 2013. p.512–519.

HINCKLEY, K.; WIGDOR, D. Input technologies and techniques. **The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications**, [S.l.], p.151–168, 2012.

JACOB, R. J. Eye movement-based human-computer interaction techniques: toward non-command interfaces. **Advances in human-computer interaction**, [S.l.], v.4, p.151–190, 1993.

JERALD, J. **The VR book**: human-centered design for virtual reality. [S.l.]: Morgan & Claypool, 2015.

LEE, J. H. et al. Projective Windows: arranging windows in space using projective geometry. In: ADJUNCT PUBLICATION OF THE 30TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY. **Anais...** [S.l.: s.n.], 2017. p.169–171.

LEWIS, J. R.; SAURO, J. The factor structure of the system usability scale. In: INTERNATIONAL CONFERENCE ON HUMAN CENTERED DESIGN. **Anais...** [S.l.: s.n.], 2009. p.94–103.

LIN, H.-C. K.; CHEN, M.-C.; CHANG, C.-K. Assessing the effectiveness of learning solid geometry by using an augmented reality-assisted learning system. **Interactive Learning Environments**, [S.l.], v.23, n.6, p.799–810, 2015.

LIU, M.; NANCEL, M.; VOGEL, D. Gunslinger: subtle arms-down mid-air interaction. In: ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE & TECHNOLOGY, 28. **Proceedings...** [S.l.: s.n.], 2015. p.63–71.

LLORÉNS, R. et al. Effectiveness, usability, and cost-benefit of a virtual reality–based telerehabilitation program for balance recovery after stroke: a randomized controlled trial. **Archives of physical medicine and rehabilitation**, [S.l.], v.96, n.3, p.418–425, 2015.

LOU, X. et al. Optimising Free Hand Selection in Large Displays by Adapting to User's Physical Movements. In: SYMPOSIUM ON SPATIAL USER INTERACTION, 2016. **Proceedings...** [S.l.: s.n.], 2016. p.23–31.

MATULIC, F.; VOGEL, D. Multiray: multi-finger raycasting for large displays. In: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2018. **Proceedings...** [S.l.: s.n.], 2018. p.245.

MELDRUM, D. et al. Virtual reality rehabilitation of balance: assessment of the usability of the nintendo wii® fit plus. **Disability and rehabilitation: assistive technology**, [S.l.], v.7, n.3, p.205–210, 2012.

MINE, M. R. Virtual environment interaction techniques. **UNC Chapel Hill CS Dept**, [S.l.], 1995.

MYERS, B. A. et al. Interacting at a distance: measuring the performance of laser pointers and other devices. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS. **Proceedings...** [S.l.: s.n.], 2002. p.33–40.

NANCEL, M. et al. Mid-air pointing on ultra-walls. **ACM Transactions on Computer-Human Interaction (TOCHI)**, [S.l.], v.22, n.5, p.21, 2015.

PEDERSON, T. Egocentric interaction. In: WORKSHOP ON WHAT IS THE NEXT GENERATION OF HUMAN-COMPUTER INTERACTION. **Anais...** [S.l.: s.n.], 2006. p.22–23.

PÖPPEL, E.; HARVEY, L. O. Light-difference threshold and subjective brightness in the periphery of the visual field. **Psychologische Forschung**, [S.l.], v.36, n.2, p.145–161, 1973.

POUPYREV, I. et al. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In: COMPUTER GRAPHICS FORUM. **Anais...** [S.l.: s.n.], 1998. v.17, n.3, p.41–52.

SANTOS, M. E. C. et al. A usability scale for handheld augmented reality. In: ACM SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, 20. **Proceedings...** [S.l.: s.n.], 2014. p.167–176.

SANTOS, M. E. C. et al. Toward standard usability questionnaires for handheld augmented reality. **IEEE computer graphics and applications**, [S.l.], v.35, n.5, p.66–75, 2015.

SIDDHPURIA, S. et al. Pointing at a Distance with Everyday Smart Devices. In: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2018. **Proceedings...** [S.l.: s.n.], 2018. p.173.

TUNG, Y.-C. et al. User-defined game input for smart glasses in public space. In: ANNUAL ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 33. **Proceedings...** [S.l.: s.n.], 2015. p.3327–3336.

VOGEL, D.; BALAKRISHNAN, R. Distant freehand pointing and clicking on very large, high resolution displays. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 18. **Proceedings...** [S.l.: s.n.], 2005. p.33–42.