FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Object Manipulation in DeskVR

Diogo Henrique Pinto de Almeida

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Daniel Filipe Martins Tavares Mendes

Second Supervisor: Prof. Rui Pedro Amaral Rodrigues

July 26, 2023

Object Manipulation in DeskVR

Diogo Henrique Pinto de Almeida

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Alexandre Miguel Barbosa Valle de Carvalho
External Examiner: Prof. Paulo Miguel de Jesus Dias
Supervisor: Prof. Daniel Filipe Martins Tavares Mendes

July 26, 2023

Abstract

Virtual reality (VR) has the potential to significantly boost the productivity of some professional settings, especially those that can benefit from immersive environments that allow a better and more thorough way of visualizing information, such as architecture, content creation, data analysis, and medicine.

A problem with VR resides in the difficulty of utilizing it for extended periods due to the usual physical demands of its mid-air movements. Recognizing this issue, DeskVR presents a solution that allows users to engage in VR while seated at a desk, integrating it into their workflow while minimizing physical exhaustion. Nevertheless, developing appropriate motion techniques for this particular context poses a challenge due to limited physical mobility and space constraints.

In this work, we specifically focused on object manipulation techniques, exploring the existing touch-based and mid-air-based approaches to gather the characteristics necessary to design a solution suitable for DeskVR that can complete object manipulation tasks of any complexity. We hypothesized that touch-based object manipulation techniques could be as effective as mid-air object manipulation in a DeskVR scenario while being less physically demanding. Thus, we proposed Scaled Indirect Touch 6-DOF, or SIT6, an indirect touch-based object manipulation technique that also incorporates scaled input mapping to address precision and out-of-reach manipulation issues. The implementation of our solution consisted of a state machine with several error-handling mechanisms and visual indicators to enhance and facilitate interaction.

We carried out user experiments to compare our solution with a baseline mid-air approach in terms of efficiency and, most importantly, physical demand and effectiveness. The results indicated that while our SIT6 technique may be slower in most object manipulation tasks, it consistently demonstrated comparable effectiveness while demanding less physical exertion. With these findings, we validated our initial hypothesis and established our proposed technique as a viable option for object manipulation in DeskVR scenarios.

Keywords: Virtual Reality, Object Manipulation, DeskVR

ACM Classification:

- Human-centered computing → Human computer interaction (HCI) → Interaction techniques → Gestural input
- Human-centered computing → Human computer interaction (HCI) → Interaction paradigms → Virtual reality

Resumo

A realidade virtual (RV) tem o potencial de aumentar significativamente a produtividade de alguns contextos profissionais, especialmente aqueles que podem beneficiar de ambientes imersivos que permitem uma melhor e mais completa visualização de informação, como a arquitetura, a criação de conteúdos, a análise de dados e a medicina.

Um dos problemas da RV reside na dificuldade de a utilizar durante períodos prolongados devido às exigências físicas habituais dos seus movimentos no ar. Reconhecendo este problema, o DeskVR apresenta uma solução que permite aos utilizadores utilizarem RV sentados numa secretária, integrando-a no seu fluxo de trabalho e minimizando a exaustão física. No entanto, o desenvolvimento de técnicas de movimento apropriadas para este contexto específico representa um desafio devido à mobilidade física limitada e às restrições de espaço.

Neste trabalho, concentrámo-nos especificamente em técnicas de manipulação de objectos, explorando as abordagens existentes baseadas no toque e no ar para reunir as características necessárias para conceber uma solução adequada para DeskVR que possa completar tarefas de manipulação de objectos de qualquer complexidade. A nossa hipótese afirma que as técnicas de manipulação de objectos baseadas no toque poderão ser tão eficazes como a manipulação de objectos no ar num cenário DeskVR, sendo menos exigentes do ponto de vista físico. Assim, propusemos o Scaled Indirect Touch 6-DOF, ou SIT6, uma técnica de manipulação de objectos baseada no toque indireto que também incorpora mapeamento escalado de entrada para resolver problemas de precisão e de manipulação fora de alcance. A implementação da nossa solução consistiu numa máquina de estados com vários mecanismos de tratamento de erros e indicadores visuais para melhorar e facilitar a interação.

Realizámos experiências com utilizadores para comparar a nossa solução com uma abordagem baseada no ar em termos de eficiência e, mais importante, de exigência física e eficácia. Os resultados indicaram que, embora a nossa técnica SIT6 possa ser mais lenta na maioria das tarefas de manipulação de objectos, demonstrou consistentemente uma eficácia comparável, exigindo menos esforço físico. Com estes resultados, validámos a nossa hipótese inicial e estabelecemos a nossa técnica proposta como uma opção viável para a manipulação de objectos em cenários DeskVR.

Palavras-chave: Realidade Virtual, Manipulação de Objetos, DeskVR

Classificação ACM:

- Computação centrada em humanos → Interação humano-computador (IHC) → Técnicas de interação → Entrada gestural
- Computação centrada em humanos → Interação humano-computador (IHC) → Paradigmas de interação → Realidade virtual

Acknowledgements

This dissertation has marked a significant milestone in my academic journey, and its completion would not have been possible without the support, incentive, and assistance of numerous people who have played pivotal roles in my success.

First and foremost, I want to express my deepest thanks to my dissertation supervisors, Professors Daniel Mendes and Rui Rodrigues, for their exceptional expertise and guidance during this work. Their unwavering availability to address any questions and the regular weekly meetings were crucial in providing me with continuous feedback and advice.

To my lab colleagues, who have accompanied me throughout this project, thank you for your ideas and advice, which have played a crucial role in this work's development. I am particularly grateful to my friend Carlos Lousada, who generously shared his VR equipment and workspace with me, along with the invaluable help provided.

I am also forever grateful to my friends, who have stood by me during this journey. Whether we met during classes, praxe, or Erasmus, you are undoubtedly the most cherished part of my university years. I want to particularly thank my close friend group Auntdulce for our incredible adventures and regular late-night Discord calls. Your friendship means the world to me.

I also want to express my heartfelt gratitude to my girlfriend for her love, support, and tireless motivation. Even though you entered my life more recently, the positive impact your presence and company have already had on me is immeasurable and everlasting.

Last but not least, to my family, particularly my parents and little brother, I am deeply indebted to you for your constant aid, love, and encouragement not just during this project but throughout my entire life. I acknowledge that I owe my accomplishments to many people, but above all, to you. Thank you for always standing by my side, believing in my dreams, and creating the conditions that allowed my pursuit of them.

Diogo Almeida

”If one does not fail at times, then one has not challenged himself.”

Ferdinand Porsche

Contents

List of Figures

List of Tables

Abbreviations

3D	3 Dimensions / 3-Dimensional
DH	Dominant Hand
DOF	Degree of Freedom
HMD	Head-mounted Display
IVE	Immersive Virtual Environment
TRS	Translate-Rotate-Scale
VR	Virtual Reality

Chapter 1

Introduction

The rapid pace of technological advancements has permanently changed how we interact with the world around us. In this era of constant innovation, one technology has notably emerged as a compelling force: Virtual Reality (VR). Using 3D head-mounted displays and sophisticated pose-tracking sensors, VR places users into fully immersive digital environments, presenting a significantly different way to engage with information.

While VR was once limited by its high cost and complexity, recent investments by industry leaders have pushed its accessibility to new heights. Now, VR equipment is more affordable, with advancements consolidating multiple components into one device, often as a single head-mounted display (HMD). As a result of its growth, the technology that had already captured the interest of industries like gaming and entertainment due to its promise of immersive experiences now sees itself with potential use cases in other diverse fields such as education and healthcare.

1.1 Context and Motivation

Virtual Reality offers its users unique capabilities by immersing them in a realistic and detailed environment. It allows physical-like interactions with virtual entities, encouraging users to use natural gestures for object selection and manipulation. Additionally, it presents a better and more thorough way of visualizing information by allowing the user to move freely around it in a 3D setting. These advantages of being within a virtual environment, which are impossible to obtain in a regular desktop experience, might help ease professional work. As such, they mainly benefit jobs that require either interaction with 3D content, such as architecture and content creation, or that demand a comprehensive look at information, which is the case for data analysis and medicine [33].

However, VR often requires tiring and extensive movements to function due to many applications requiring mid-air movements similar to natural gestures, making its use hard in work environments for prolonged periods. With this in mind, DeskVR comes as a solution; it allows users to be fully immersed in Virtual Reality while sitting at an office desk without needing exhausting and prolonged movements [40]. Therefore it seamlessly integrates a virtual environment into a user's workflow and workplace, potentially increasing productivity. Considering this and the many advancements in virtual reality technology for DeskVR, in both price and performance, the potential for this concept to become a viable option for professional work settings might already be present.

1.2 Challenges

Since many existing VR techniques are primarily designed for users in a standing position and rely on physically demanding mid-air movements for interaction, it is challenging to find practical VR solutions for a seated context. Hence, the requirements of DeskVR demand the exploration of alternative approaches that enable comfortable and efficient movement, object selection, and object manipulation within the virtual space.

The complexity of developing such techniques arises from the challenge of achieving natural and intuitive interactions while being imposed by the limitations of a seated position. Unlike standing-based VR experiences, DeskVR users have restricted physical mobility and may have limited space to perform large-scale movements. Thus, techniques must be carefully designed, considering the range and intensity of movements required and the ergonomics of the user's seated position.

Ultimately, these techniques must integrate seamlessly into the user's workflow, providing an immersive experience without requiring extensive physical effort. It is also necessary that the approaches' gesture mappings maintain a straightforward, user-friendly interface that can still provide all the necessary functionality.

1.3 Objectives

This dissertation focuses specifically on the manipulation of 3D objects in DeskVR, diverging this work from the selection of objects since it has already been studied previously [22]. Thus, we aimed to propose a novel technique composed of undemanding and intuitive gestures for the translation and rotation of objects while in a seated position. With this approach, we also sought to tackle several problems in virtual object manipulation, such as precision issues and out-of-reach manipulation.

To achieve this, we extensively explored different object manipulation techniques, with a particular emphasis on touch-based approaches and their corresponding gesture mappings. We believe that these techniques are better suited for DeskVR than mid-air approaches, as they require minimal movement and can be executed effectively while in a seated position. Additionally, we thoroughly examined mid-air techniques that specifically addressed precision and out-of-reach manipulation challenges. Our analysis focused on studying their input mapping strategies, aiming to adapt and integrate them into our touch-based solution.

1.4 Hypothesis and Research Questions

Considering the previously specified problem and the established objectives, this work puts forward the following hypothesis:

Touch-based object manipulation techniques can be as effective as mid-air object manipulation in a DeskVR scenario while being less physically demanding.

While existing touch-based object manipulation techniques were not originally designed with DeskVR in mind, some of them could possibly be adapted to this scenario. The key lies in identifying touch-based techniques incorporating inherently intuitive gestures for object manipulation that work in a seated-desk context. By adapting the motion dictionaries inherent to these approaches and combining them with efficient input mapping, it might be possible to achieve task effectiveness comparable to that of mid-air techniques. With this in mind, we propose Scaled Indirect Touch 6-DOF (SIT6), a touch-based solution incorporating these elements.

Thus, the aforementioned hypothesis can be divided into the following research questions:

- **RQ1**: Can SIT6 have an equal or greater success rate at object manipulation tasks compared to a state-of-the-art mid-air baseline with manipulation capabilities for any distance?
- **RQ2**: Can SIT6 be faster at completing object manipulation tasks than a state-of-the-art mid-air baseline with manipulation capabilities for any distance?
- **RQ3**: Can SIT6 complete object manipulation tasks while being less physically demanding than a state-of-the-art mid-air baseline with manipulation capabilities for any distance?

1.5 Document Structure

Chapter 2, titled *Related Work*, provides an in-depth analysis of previous research articles encompassing the topics studied in this work. This chapter analyzes touch-based and mid-air object manipulation techniques within virtual environments while exploring approaches that address precision issues and out-of-reach manipulation. Moreover, the chapter discusses what characteristics a DeskVR solution should contain from the studied approaches.

Chapter 3, named *Scaled Indirect Touch 6-DOF (SIT6)*, introduces our approach to the presented problem. Within this chapter, we provide a comprehensive overview of the design specifics of our proposed solution while delving into the details of each component incorporated in its implementation.

Chapter 4, *User Evaluation*, explains the experimental methodology developed to assess our solution's performance. It details the implementation of the selected baseline of comparison, the test setup and environment, the testing procedure, and the quantitive and qualitative metrics employed to measure and analyze the results obtained from the experiments.

Chapter 5, titled *Results*, shows the outcomes of the conducted user experiments. Within this chapter, the objective metrics collected during the experiments are meticulously analyzed for each task. As for the subjective data gathered, it is thoroughly examined for each question posed to the participants. Moreover, this chapter delves into a comprehensive discussion of the obtained results, providing insights into their significance and implications.

Lastly, Chapter 6, *Conclusions*, presents the key findings from the preceding chapters and offers the concluding remarks on this work. Additionally, it presents possibilities for future research to explore the addressed topic further.

Chapter 2

Related Work

In this chapter, we explore several papers regarding the existing object manipulation techniques inside virtual environments. For the classification of the various methods, we follow a taxonomy similar to the one defined in a survey by Mendes et al. [20], in which they presented, reviewed, and discussed several object manipulation techniques.

Thus, we begin by reviewing the literature on touch-based techniques, analyzing the different gesture dictionaries for direct, indirect, and widget-based approaches. Subsequently, we delve into research that explores mid-air manipulation methods, focusing on input mapping strategies that address precision and out-of-reach issues. Finally, after thoroughly investigating the relevant approaches, we discuss the key characteristics that render a technique suitable for DeskVR, enabling us to design a solution that incorporates those desirable attributes.

2.1 Touch-Based Interactions

Over the years, several 3D object manipulation techniques based on multi-touch interactions have been proposed and evaluated, with efforts being made to create content interactions that are more natural and can effectively outperform mouse-based inputs [15].

For classification purposes regarding DeskVR suitability, we divided the touch-based object manipulation techniques into three distinct categories:

- **Direct Touch Techniques**, which group the methods that require directly touching the object through the display to enable manipulations;

- **Widget-Based Techniques**, which define the approaches that use virtual widgets and require directly touching them to perform manipulations;

- **Indirect Touch Techniques**, which gather the techniques that can be performed through an external touch surface, therefore not needing to touch the object directly for manipulations.

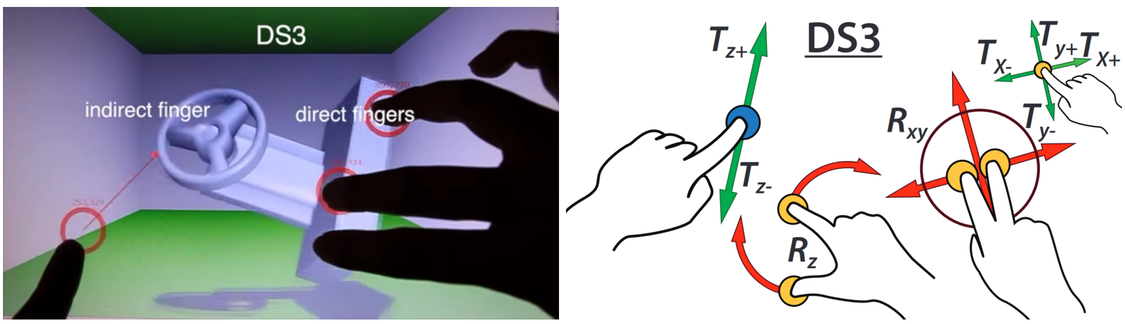
2.1.1 Direct Touch Techniques

Since research suggests that rotation and translation are not separable in the human mind [37], studies initially proposed object manipulation approaches that could control multiple DOFs simultaneously. With this in mind, Hancock et al. initially proposed several methods to manipulate 6 DOFs at once using one to three touches [12]. These approaches were then improved with Sticky Fingers & Opposable Thumb [13], which work by keeping the user's fingers in touch with the virtual object in the position they initially reach. This technique gives the perception of touching the virtual object by providing some feedback one might expect in the physical world.

Reisman et al. [26] also proposed a technique that can control multiple DOFs simultaneously. This solution consists of a screen-space formulation for manipulating 3D objects in 6 DOFs using multiple contact points on a multi-touch device. The touch points remain constant during the interaction, and a constraint solver moves and rotates the objects simultaneously.

However, solutions that can execute distinct transformations simultaneously often cause unintentional operations to occur. Therefore, several approaches with DOF separation were proposed. Martinet et al. [19] designed DS3 (Fig. 2.1), which uses one touch to move the object in the screen plane and an indirect touch to manipulate the object's depth, with two direct touches in the object enabling rotation. The authors concluded, after comparing the techniques to other existing approaches [13] [26], that separating translation DOF from rotation DOF led to the best performance.

Another approach that separates DOFs was proposed by Liu et al. [17], which, instead of using the number of touches to determine the type of transformation to apply, uses the movement properties of two touches. Two moving touches control 3 translation DOFs and 1 rotation DOF; one fixed touch and another moving touch control the remaining 2 rotation DOFs. According to the authors, this technique outperforms DS3 and Screen Space; however, it might not be satisfactory when precise control of object transformations is required.



(a) (b)

Figure 2.1: DS3 [19]: (a) The user manipulates the object's depth using an indirect touch. (b) The technique's gesture dictionary (taken from [31]).

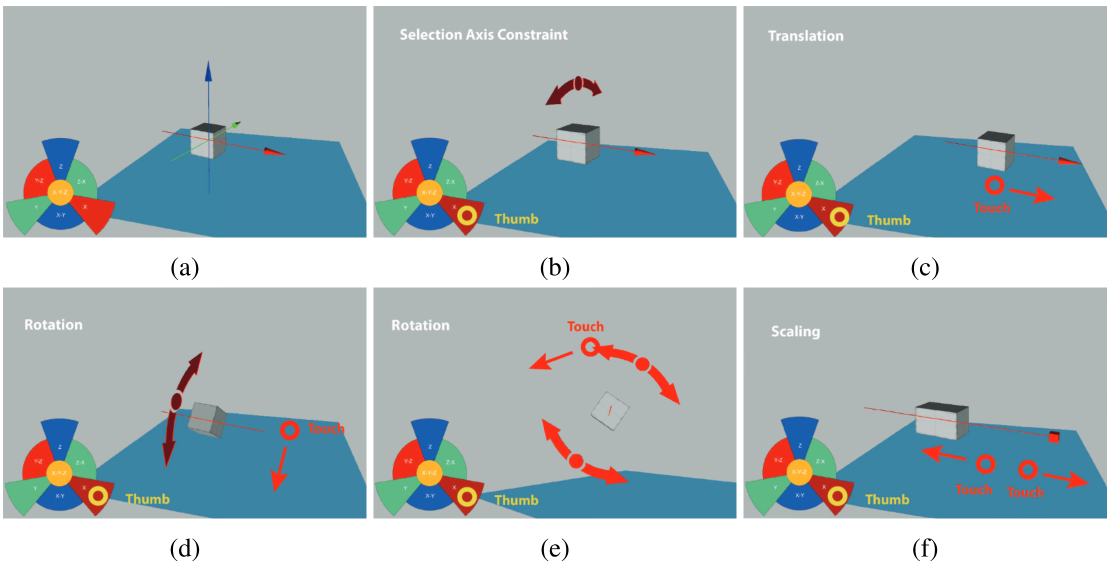
2.1.2 Widget-Based Techniques

One of the approaches to input remapping is using widgets, which is common in mouse-based methods such as Arcball [30]. Techniques that use widgets can have very distinctive interactions, depending on the widget type.

Cohé et al. [9] proposed tBox, a 3D transformation widget consisting of a wire-frame cube in which users can drag one of its edges to move the object and move one of its faces to perform rotations. A similar approach is made with GimbalBox [4], which also uses a box-shaped widget around the object; by touching one of the box's faces, the user induces a translation, with rotations being performed by either using TRS or touching one of the edges of the box.

Widgets can also be implemented outside the object, as seen with TouchSketch [39] (Fig. 2.2), which resorts to a constraint menu that divides manipulations into three categories: axis-constrained manipulation, plane-constrained manipulation, and uniform manipulations. The user's non-dominant hand can choose a constraint from the menu, while the dominant hand performs transformations based on the set constraint.

Although widget-based techniques necessarily require the existence of a widget and direct interaction with it, some approaches also utilize direct touches with the object. For instance, Mendes et al. [21] proposed LTouchIt, which employs this mixed approach, using direct object touches for translations and widgets, in the form of virtual handles, for rotations.



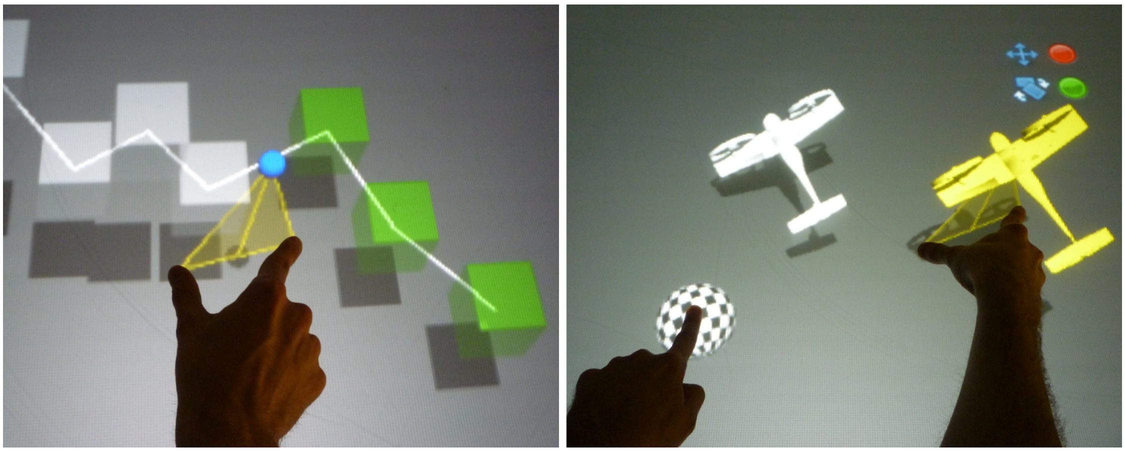
(a) (b) (c) (d) (e) (f)

Figure 2.2: TouchSketch [39]: Gestures for axis-based transformation manipulations. (a) The initial state of an object before manipulation. (b) After specifying the X-axis constraint, only the red axis is displayed. (c-f) After an axis-constraint is selected, users can translate, rotate and scale the object by using the DH.

2.1.3 Indirect Touch Techniques

Regarding an indirect touch paradigm, Au et al. [1] proposed a constraint-based technique similar to TouchSketch [39]. However, the approach depends on a single multi-touch gesture instead of relying on a menu to select constraints. Therefore, users can choose a possible axis with two touch points, either inside or outside the object, and transformations are achieved by moving and holding the two touching fingers.

Other relevant techniques in this paradigm are the ones applied to object manipulation in stereoscopic tabletops. Since objects appear outside the surface in stereoscopic environments, indirect touch interactions for manipulations are required so that the user's hand does not obstruct the projected objects.

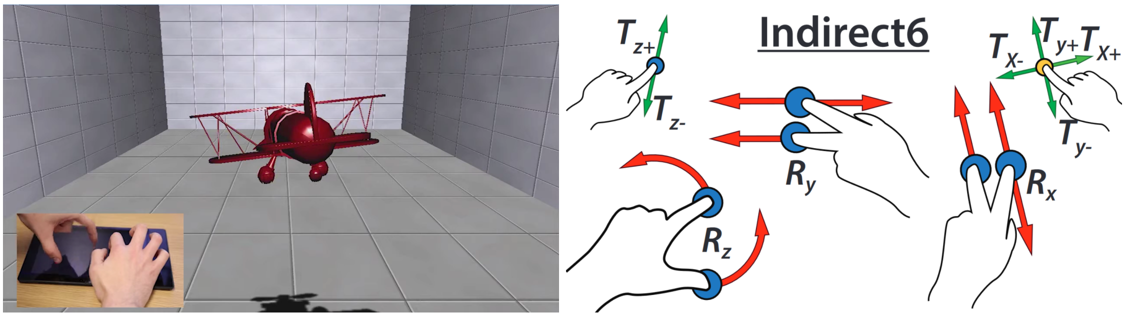


(a) (b)

Figure 2.3: Illustration of Triangle Cursor [35] in action. (a) Using a single hand, the user controls the 3D cursor's position and height above the surface. (b) Rotations are enabled with the other hand.

Benko et al. [3] proposed a technique that allows the user to control a cursor through a balloon metaphor. The user can move this cursor using one finger, which carries the object in the plane. To move the object upwards or downwards, the user moves another finger closer or farther away from the first touch point. Another form of controlling a cursor for object manipulations in a stereoscopic environment was proposed by Strothoff et al. [35] (Fig. 2.3). In this technique, when the user touches the surface at two points, a triangle is displayed with the two base vertices at the touch positions. The triangle's position can be controlled by moving the fingers on the surface. The distance between the two fingers controls the height above the surface (the triangle's third vertex). When the user touches the surface with the free hand, a trackball is displayed at the touch point, which is used to rotate the object in the remaining 2 DOFs

Alternative approaches that do not utilize stereoscopic tabletops were proposed by Simeone [31], which provide indirect touch interaction by using an external multi-touch surface. The presented techniques, Indirect4 and Indirect6 (Fig. 2.4), can control 4 DOFs and 6 DOFs, respectively. Indirect4 employs a touch from the dominant hand for horizontal movement and a touch from the non-dominant hand for either adjusting the object's vertical position or rotating around a vertical axis. Indirect6 controls the object's position similarly but can manipulate 2 additional DOFs by using two touches from the non-dominant hand to perform rotations. Horizontally moving two fingers controls yaw, while vertically moving them controls pitch. Driving the two fingers in opposite directions controls roll. The author compared these techniques to DS3 [19] and Triangle Cursor and found that indirect touch interaction methods offer a more comfortable viewing experience with no significant differences in net manipulation times.



(a) (b)

Figure 2.4: Indirect6 [31]: (a) The user performing a roll rotation. (b) The technique's gesture dictionary.

2.2 Mid-Air Interactions

Mid-air-based interactions commonly perform inputs in a spatial 3D environment, allowing them to manipulate objects with potentially more natural gesture dictionaries. This type of interaction can be achieved by either tracking the user's hands with external sensors and cameras or using tracked handheld controllers or wearable devices.

To this extent, in immersive virtual environments, object manipulation can be performed similarly to how it is performed with physical objects. Thus, this metaphor translates the user's hand into a Simple Virtual Hand [7], which is natural but insufficient in situations where precision and an extended range of translation and rotation are required.

In this section, we initially explore various approaches for mid-air object manipulation to analyze their different gesture dictionaries. Subsequently, we look into mid-air object manipulation solutions explicitly designed to mitigate out-of-reach manipulation problems or precision issues to study the different strategies used to tackle these difficulties.

2.2.1 Within Arm-Length

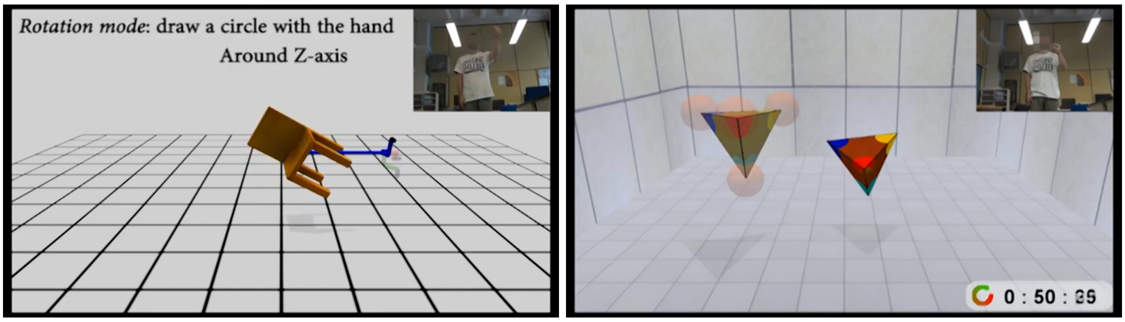
Finding techniques that effectively map user motion in the 3D space to object movements is challenging for mid-air interactions. Although the Simple Virtual Hand metaphor can perform translations adequately, it does not offer an efficient solution regarding rotation and scaling. Therefore, several authors have conducted research to propose practical solutions for mid-air manipulation metaphors.

Several approaches expand upon the interactions on a tabletop, utilizing the space above it. Hilliges et al. [14] introduced a technique that can switch between touch-based and mid-air interactions, allowing for interactions based on depth. This technique employs computer vision to track the user's hand in 4 DOFs, with the grab gesture being able to be detected. To this degree, shadows of the user's hand are cast into the scene and used to manipulate the virtual objects. Another technique that combines the use of a multi-touch surface and the space above it is Air-TRS [10]. With this approach, users can use one hand to directly grab and move an object, with the second hand allowing rotations around the first hand after performing a grab gesture outside the object.

Mapes and Moshell proposed Spindle [18], a distinct approach that uses both hands to manipulate virtual objects. The midpoint between the user's hands acts as the transformation center. Moving both hands simultaneously in the same direction translates the object, and moving them around the center rotates the object. To scale the object, the user changes the distance between both hands.

Several other techniques were adapted from Spindle. Song et al. [32] proposed a Handlebar metaphor, which tracks the position of the user's hands in space using a single depth camera, tracking them in 3 DOFs. Thus, rotations around the axis of the line defined by both hands (the handlebar) can be achieved with a single swivel gesture. The approach also allows users to manipulate multiple objects along the handlebar. Following this research, Cho et al. proposed Spindle+Wheel [8], which resorts to spherical handheld devices for hand-tracking. This two-handed approach performs translations by moving both hands in the same direction and enables rotations by moving both hands in opposite directions (roll and yaw). Scaling operations are performed by adjusting the distance between hands while rotating one hand rotates the object around the central axis of the handheld device (pitch). The main difference between the Handlebar and Spindle+Wheel techniques is that the latter can perform simultaneous 7-DOF transformations.

Bossavit et al. [5] proposed two other distinct mid-air manipulation techniques. The first one, Crank Handle, is a one-hand technique that isolates translation from rotation. The approach also decomposes the rotation into primary axes, which can be selected through a crank handle metaphor. The second technique, Grasping Object, is also a one-hand manipulation technique. However, it merges translation and rotation and does not separate rotation. Figure 2.5 presents both techniques.



(a) (b)

Figure 2.5: Illustration of both techniques [5] in action. (a) Crank Handle in rotation mode. (b) Grasping Object during a docking task.

2.2.2 Out-of-reach Manipulation

One of the challenges regarding object manipulation in immersive virtual environments is the interaction with objects that are out of the users' reach.

Several solutions have been designed to mitigate this problem, such as the Go-Go [25] technique, which uses the metaphor of interactively extending the user's arm and employs nonlinear mapping for reaching and manipulating objects at a distance. Upon detecting movement by the user's hand, a 1:1 mapping is used up until a certain distance. If the user moves their hand further, the arm extends according to a predefined coefficient. Therefore, this technique allows for smooth and direct control of close and out-of-reach objects.

Bowman et al. [6] compared mid-air techniques such as Go-Go and ray casting, concluding that these approaches had several limitations. With this evaluation in mind, they proposed HOMER (Hand-centered Object Manipulation Extending Ray-casting), a hybrid 3D manipulation technique that uses ray-casting for selection and a virtual hand for manipulation, which is placed in the object upon selection. In this approach, the distance between the user's torso and the object is directly mapped to the distance separating the user's torso and the controller (the user's physical hand) at the time of selection. This mapping can be visualized in Figure 2.6, and is calculated by the following equation:

$$D_{virthand} = D_{currhand} * \frac{D_{object}}{D_{hand}} \tag{2.1}$$

Where $D_{virthand}$ is the distance of the virtual hand from the user's body, $D_{currhand}$ is the current distance between the user's torso and hand, D_{object} is the initial distance between the user's torso and the selected object, and D_{hand} is the initial distance between the user's torso and hand.

A distinct method for out-of-reach object interaction in large virtual environments is the Worlds in Miniature technique [34]. In this approach, users control a miniature of the virtual world, allowing them to move around promptly, change their point of view, or interact with virtual objects. Another distinct method was proposed by Pierce et al. [24] with the Voodoo Dolls

[1] Image taken from the author's slides: https://slideplayer.com/slide/5336216/

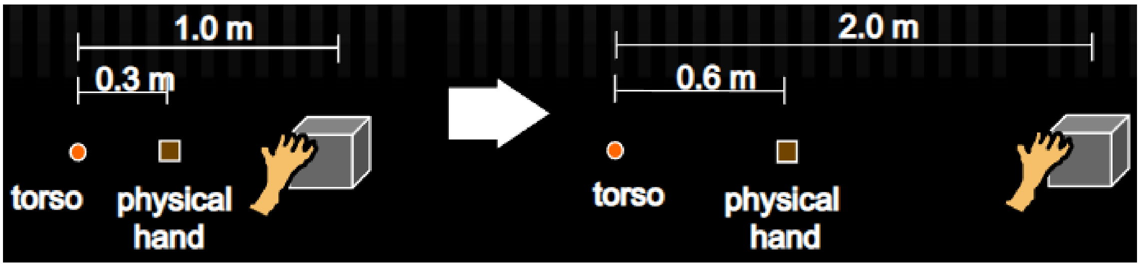


Figure 2.6: 1:N translation mapping in HOMER [6] [1].

technique, which allows the user to manipulate objects regardless of scale. A dynamic resizing method is used to produce handheld versions of the objects, referred to as "dolls," which can then be manipulated instead of the objects themselves. Therefore, this method enables users to work on objects at different scales without requiring explicit resizing of the objects or the environment.

2.2.3 Solutions for Precision Issues

Another common problem with object positioning techniques in immersive virtual environments is the need for more accuracy.

There have been solutions for this issue based on discrete placement constraints (snapping) and collision avoidance mechanisms [16]. However, Frees et al. [11] introduced PRISM (precise and rapid interaction through scaled manipulation), a technique that doesn't restrict the placement of objects. This approach, in contrast to methods such as Go-Go [25], which amplify hand movements for remote manipulation, actually scales hand movements down to enable precise control over objects, as seen in Figure 2.7. This increase of the control-to-display ratio PRISM performs results in a slower movement of the cursor or object compared to the user's hand, reducing hand jitter and creating an offset between the object and hand, which provides the user with precise control over the manipulated object's position. User evaluations by the authors showed that PRISM performs faster and is preferred by users over conventional direct methods.

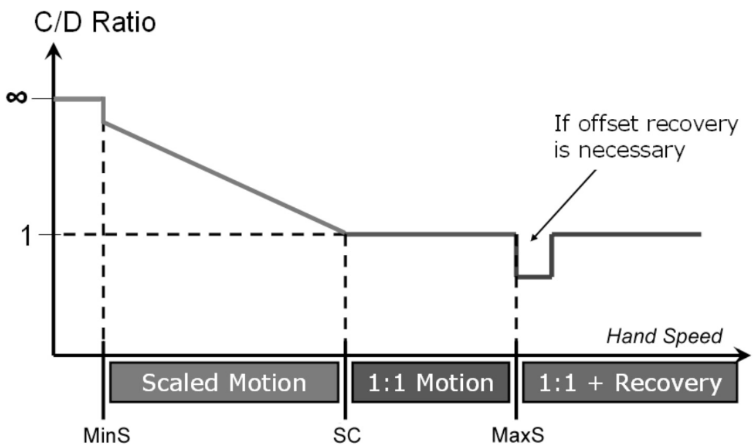


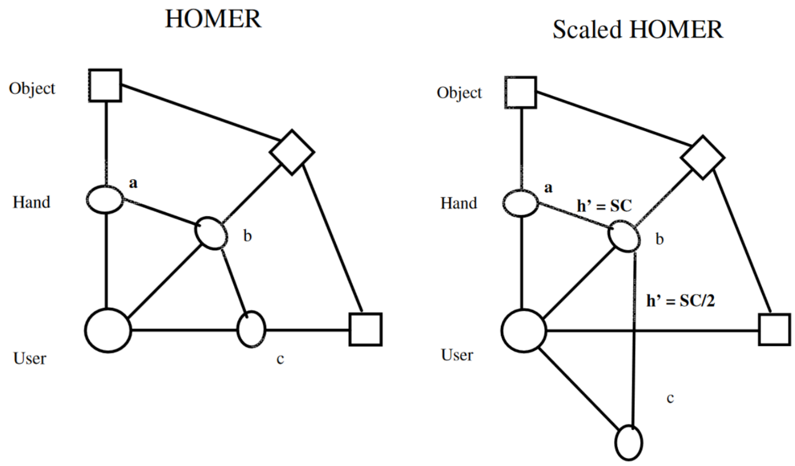
Figure 2.7: Simplified interface diagram showing how PRISM [11] uses Hand Speed to adjust CD.

Several solutions combine this precise mapping from PRISM with long-reach approaches. Wilkes et al. [38] proposed an evolution over the HOMER [6] technique: Scaled HOMER, which adds velocity-based scaling to provide more precise object control (Fig. 2.8). Thus, when the hand is moving quickly, the scaled hand distance will be equal to or greater than the actual distance; when the hand is moving slowly, the scaled hand distance will be less than the actual distance, providing fine-grained control of the object's position. The scaled distance is obtained through the following equation:

$$SD_{hand} = min(\frac{Velocity_{hand}}{SC}, 1.2) * D_{hand} \tag{2.2}$$

Where SD_{hand} is the hand's scaled distance, $Velocity_{hand}$ is the current hand velocity, SC is a predefined scaling constant, and D_{hand} is the initial distance between the user's torso and hand.

It is noteworthy that the authors capped the scaling factor ($min(\frac{Velocity_{hand}}{SC}, 1.2)$) at a maximum value of 1.2 since "pilot tests showed that scaling up more than 1.2:1 was difficult for users to handle". Furthermore, the authors found that the technique outperformed HOMER in various task scenarios, especially those that required high precision, distant object placement, or an extensive movement range.



(a) (b)

Figure 2.8: (a) Object motion in HOMER [6]. (b) Object motion in Scaled HOMER [38].

Another approach that utilizes PRISM was presented by Auteri et al. [2], combining the technique's scaling factor with Go-Go [25] to increase precision for extended reach object manipulation. The solution begins by directly applying PRISM to the movement of the user's hand, referred to as the base cursor, which calculates a new cursor position, the PRISM cursor, based on velocity-based scaling. The Go-Go distance-based heuristic then amplifies the PRISM cursor's movement. The combination of both of these techniques increased task completion success and allowed for more precise manipulation.

Some techniques make use of widgets to tackle precision issues. Nguyen et al. [23] presented an approach consisting of a triangle-shaped widget with seven points - 7-Handle. The triangle's three vertices are the first-level handles used to position and control 3 DOFs precisely. The triangle's three midpoint side handles, called second-level handles, control the two adjacent first-level handles. Finally, the third-level handle, the centroid of the three first-level handles, can be used to manipulate 6 DOFs directly.

2.3 Discussion

Throughout this chapter, we have explored several touch-enabled and mid-air techniques, presenting their solutions to the challenges of object manipulation. We have also shown the unique features of these approaches and how their gesture dictionaries performed regarding usability and task completion.

While the object manipulation techniques presented earlier were not designed for DeskVR, some possess specific characteristics that make them suitable in a seated-position scenario. In this section, we start by separating the techniques by their interaction paradigm, touch-based or mid-air-based, and then compare their characteristics, analyzing their suitability for a DeskVR context.

Regarding the analyzed characteristics, we will study the following:

- For both paradigms of interaction:
  - **Number of controlled DOFs**, that refers to the number of directions in which a manipulation technique can move an object;
  - **Reach**, which represents how far from the user the object manipulation can be performed;
  - **Mapping**, that refers to how a user's input translates onto actions performed on the manipulated object.
- For touch-based interactions, specifically:
  - **Type of contact**, which can be direct, indirect, or through a widget depending on whether or not there is direct contact with an object or widget.
- For mid-air interactions, specifically:
  - **Tackled issue**, that refers to whether or not the technique represents a solution for out-of-reach or precision issues.

Table 2.1 summarizes the analyzed characteristics in this section.

2.3.1 Touch-Based Interactions

Regarding the type of contact for touch-based approaches, techniques that utilize direct touch usually rely on touch-enabled displays [13, 19, 17] since the user's hand has to be in direct contact with the shown object. The same applies to widget-based techniques, which usually require direct contact with the widget on a touchscreen [9, 21, 4, 39]. However, there are cases of widget-based approaches that, although implemented on a touch display, do not require a direct touch on the widget, therefore being able to be implemented in an external touch surface [1]. Finally, methods that rely on indirect touch interact with objects through either an external touch surface or a stereoscopic tabletop [3, 35, 31].

After analyzing each type of contact and the techniques within them, we can conclude that both the direct touch and widget-based approaches are unsuitable for DeskVR since it is impossible to touch an object or widget within a VR environment directly. On the other hand, it is trivial to implement indirect touch techniques in VR since the inputs are performed outside the object. Additionally, an external touch frame can reasonably emulate a desk surface, which means that the approaches that employ this input device fit the DeskVR experience well.

For the number of controlled DOFs, all the analyzed techniques allow translations for all 3 DOFs, with most of the approaches having 3 DOFs for rotation [13, 19, 17, 21, 1, 39, 31]. However, some methods limit rotations to 1 DOF [9, 4, 3, 35]. Furthermore, only widget-based techniques allow scaling operations, with some exclusively allowing 1 DOF [1] and others allowing 3 DOFs [9, 39].

Having 3 DOFs for translation and rotation is essential for the seated-desk scenario. Thus, only gesture dictionaries from approaches that allow 6 DOFs are suitable, with the possibility of expanding the motion set for specific 4 DOFs techniques to remove rotation limitations, which is the case for Triangle Cursor [35]. Since only widget-based approaches allow scaling operations, designing a technique with a widget for scaling is possibly one of the ways of implementing a solution with at least 7 DOFs in a DeskVR context. However, as seen previously, object manipulation in VR requires an indirect touch paradigm, making it challenging to interact with widgets and perform the scaling operation. Finally, DOF separation would also be preferred to avoid unintentional operations, which are more likely to happen in VR since users cannot see their movements.

As for reach, techniques are classified as having screen-space reach when the environment is displayed in a traditional non-stereo screen. These approaches allow users to manipulate an object, independently of how far it is, within the limits of the screen [13, 19, 17, 9, 21, 1, 4, 39]. Arm-length reach refers to techniques where the length of the user's arm limits where the user can move objects for manipulation [3, 35, 31]. Additionally, we have scaled and infinite reach approaches within the arm-length techniques category, which will be explored in the following subsection, along with mid-air techniques.

Screen-space techniques are not suitable for a DeskVR scenario since the user is not restricted to interacting with objects within the confines of a screen in this environment. Instead, a seated virtual reality experience would require arm-length reach techniques, allowing the implementation of a solution on an external touch surface, where only the user's arm length limits the range of object motion.

2.3.2 Mid-Air Interactions

We begin by analyzing reach for mid-air techniques, which can be arm-length, scaled, or infinite. As mentioned previously for touch-based methods, arm-length reach refers to approaches where the length of the user's arm defines the distance range of manipulations [7, 14, 10, 32, 8, 5, 23]. Scaled reach techniques are where the extent of the users' sweep while moving an object is more significant than its arm's length but can not reach infinite. This scaling effect is achieved through scaled input mapping [11, 25, 6, 38]. As for infinite reach techniques, these group the methods with no limit to where objects can be so that users can manipulate them [34, 24].

We have seen previously that for touch-based techniques, a solution for a DeskVR scenario should utilize arm-length reach. However, we can now conclude from this analysis that an approach with scaled reach would also be possible and, with the assistance of its input mapping, it would aid in tackling several object manipulation technique problems.

As for mapping, an exact manipulation maps the movement of a device, or a hand tracked directly onto the virtual object transform, offering a 1:1 control [7, 14, 10, 32, 8, 5]. On the other hand, a scaled manipulation maps the input using a linear or nonlinear scaled transform to improve accuracy or increase the range of transform parameters through N:1 or 1:N controls, respectively. [11, 25, 6, 38]. A hybrid manipulation applies different mappings to different DOFs of the same transformation. Furthermore, we have remapped manipulation, which maps tracked DOFs onto different manipulation DOFs or uses other input channels to control object transform DOFs [23, 34, 24].

While we could implement all of the manipulation mappings in an approach for a seated-desk scenario, it would be beneficial, as mentioned earlier, to take advantage of the scaled mappings of some techniques (e.g., Scaled HOMER [38]). This scaled mapping would help design a solution that could tackle precision and out-of-reach issues. When the user wants to move an object precisely, the object should move slower than the user's hands [11]. Alternatively, precision may not be a priority when relocating an object from one remote location to another, and the user may move relatively quickly [6].

2.3.3 Conclusions

After analyzing the characteristics of state-of-the-art touch-based and mid-air techniques, we now need to examine which of these paradigms of interaction is the most fitting for DeskVR. Touch-based techniques do not usually require tiring motion gesture-wise since most rely on simple touch inputs, reinforcing their suitability for a seated desk context. As for mid-air techniques, these approaches do not particularly fit the seated desk experience since they often demand exhausting movements or even require the user to stand up.

Considering this and the previous analysis, we can conclude that a suitable object manipulation approach for a seated-desk environment that could tackle precision and out-of-reach problems would have to have the following characteristics:

- **Paradigm of Interaction**: Touch-based;
- **Type of Contact**: Indirect;
- **Available DOFs**: At least 6 (3 for Translation + 3 for Rotation) with separation;
- **Reach**: Scaled;
- **Mapping**: Scaled N:1 and Scaled 1:N to tackle both precision and out-of-reach issues, respectively.

Coming up with an approach with all these conditions is challenging; however, by gathering and adapting the gesture dictionaries and features from the analyzed techniques, we can design a solution that effectively solves object manipulation problems and is fitting for DeskVR.

Table 2.1: Classification of the main object manipulation techniques.

Technique	Paradigm of Interaction	Translation DOFs	Rotation DOFs	Scaling DOFs	Reach	Mapping	Type of Contact	Tackled Issue
Sticky Fingers [13]	Touch-Based	3	3	0	Screen-Space	Exact	Direct	N/A
DS3 [19]	Touch-Based	3	3	0	Screen-Space	Hybrid	Direct	N/A
Liu et al. [17]	Touch-Based	3	3	0	Screen-Space	Exact	Direct	N/A
tBox [9]	Touch-Based	3	1	3	Screen-Space	Exact	Widget	N/A
GimbalBox [4]	Touch-Based	3	1	0	Screen-Space	Remapped	Widget	N/A
TouchSketch [39]	Touch-Based	3	3	3	Screen-Space	Remapped	Widget	N/A
LTouchIt [21]	Touch-Based	3	3	0	Screen-Space	Exact	Widget	N/A
Au et al. [1]	Touch-Based	3	3	1	Screen-Space	Remapped	Indirect	N/A
Balloon Selection [3]	Touch-Based	3	1	0	Arm-Length	Hybrid	Indirect	N/A
Triangle Cursor [35]	Touch-Based	3	1	0	Arm-Length	Hybrid	Indirect	N/A
Indirect6 [31]	Touch-Based	3	3	0	Arm-Length	Hybrid	Indirect	N/A
Simple Virtual Hand [7]	Mid-Air	3	3	0	Arm-Length	Exact	N/A	N/A
In the Air [14]	Mid-Air	3	1	0	Arm-Length	Exact	N/A	N/A
Air-TRS [10]	Mid-Air	3	3	1	Arm-Length	Exact	N/A	N/A
Handle Bar [32]	Mid-Air	3	3	1	Arm-Length	Exact	N/A	N/A
Spindle + Wheel [8]	Mid-Air	3	3	1	Arm-Length	Exact	N/A	N/A
Crank Handle [5]	Mid-Air	3	3	0	Arm-Length	Exact	N/A	N/A
Grasping Object [5]	Mid-Air	3	3	0	Arm-Length	Exact	N/A	N/A
PRISM [11]	Mid-Air	3	3	0	Arm-Length	Scaled N:1	N/A	Precision
7 Handle [23]	Mid-Air	3	3	0	Arm-Length	Remapped	N/A	Precision
Go-Go [25]	Mid-Air	3	3	0	Scaled	Scaled 1:N	N/A	Out-of-Reach
HOMER [6]	Mid-Air	3	3	0	Scaled	Scaled 1:N	N/A	Out-of-Reach
Worlds in Miniature [34]	Mid-Air	3	3	0	Infinite	Remapped	N/A	Out-of-Reach
Voodoo Dolls [24]	Mid-Air	3	3	0	Infinite	Remapped	N/A	Out-of-Reach
Scaled HOMER [38]	Mid-Air	3	3	0	Scaled	Scaled N:1/1:N	N/A	Precision + Out-of-Reach

Chapter 3

Scaled Indirect Touch 6-DOF (SIT6)

In light of the lack of object manipulation techniques designed for a DeskVR environment, our objective was to develop a viable solution tailored to this setting that could also effectively handle precise and out-to-reach object placements. Consequently, drawing inspiration from the examined approaches outlined in chapter 2, we created Scaled Indirect Touch 6-DOF, or SIT6, an indirect touch object manipulation technique with distance and velocity-based scaling.

3.1 Design

Our solution incorporates a gesture dictionary adapted from Indirect6 [31]. Hence, it offers the advantages of touch-based interaction, indirect manipulation, and 6 DOF. However, the original design of Indirect6 primarily targeted generic displays and did not consider the specific demands of virtual reality applications. To address this limitation, we have modified the user's gesture mappings on the touch surface for our technique.

Moreover, to ensure that our proposed approach has scaled reach and mapping of both scaled N:1 and 1:N, effectively addressing the issues related to precision and out-of-reach interactions, we will employ a distance and velocity-based mapping approach inspired by the principles of previously studied approaches such as PRISM [11] and Scaled HOMER [38].

3.1.1 Gesture Dictionary

Unlike in the original design of Indirect6, one-finger gestures in SIT6 translate the object across the XZ-plane instead of in the XY-plane. To translate the object along the Y-axis, the user applies a second touch, moving the finger forward and backward on the touch surface to translate the object up and down. Thus, the translation motions stand comparable to those of Balloon Selection [3].

By swapping the Z-axis translation and Y-axis translation gestures relative to Indirect6, our goal is to enhance the intuitiveness and natural feel of the input by aligning the object's movement with the same plane as the user's hand rather than with the screen's plane, ensuring that the user's gestures have the intended effect in a three-dimensional VR space and aligning with the immersive nature of VR technology.

In addition to the translation gestures change, the rotation motions in SIT6 have also undergone modifications, with the gestures for yaw (Ry) and roll (Rz) having been switched. This adjustment has the same goal as the translation gestures change, with the rotation movement following the user's hand in the same plane. Lastly, the rotation gestures in SIT6 have been optimized to require only two touches instead of three, thus requiring one less hand to be in contact with the touch surface in most cases.

Figure 3.1 illustrates the solution's gesture dictionary, which incorporates all these modifications. The gestures required to engage in each type of transformation are the following:

- **XZ-plane Translation**: Movement in any direction with a single touch (Figs. 3.1a, 3.1c);
- **Y-axis Translation**: Vertical movement with one touch and an additional stationary touch (Fig. 3.1b);
- **X-axis Rotation (Pitch)**: Vertical movement with two touches (Fig. 3.1d);
- **Y-axis Rotation (Yaw)**: Circular motion with two touches (Fig. 3.1e);
- **Z-axis Rotation (Roll)**: Horizontal movement with two touches (Fig. 3.1f).

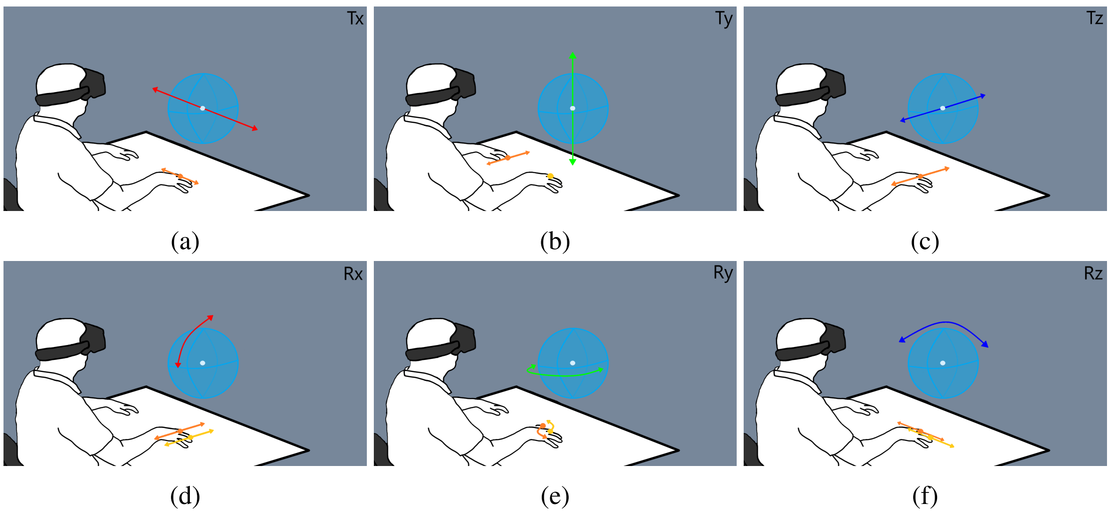


Figure 3.1: Gesture dictionary for the proposed solution. The image's silhouette was adapted from Sousa et al. [33].

3.1.2 Input Mapping

For SIT6, we adapted the mapping approach from Scaled HOMER, combining distance-based and velocity-based mappings to facilitate precision and out-of-reach placements. For the distance-based scaling implementation, we faced challenges in accurately estimating the distance between the user's hand and torso within a touch-based context. As a result, we relied exclusively on the distance between the user and the object in the virtual environment, multiplied by a predefined coefficient, for the calculation process. This adaptation effectively scales object interactions at any distance, despite the limitations in directly measuring the hand-to-torso distance.

Regarding the computation of velocity-based scaling, we employ the finger velocity divided by a scaling constant. Our implementation also limits the value of this scaling factor to 1.2. This upper limit is set because, similarly to Scaled HOMER, it becomes challenging for users to control the object when using higher values. By capping the scaling factor, we aim to achieve a balance between allowing efficient object manipulation and maintaining user control within manageable limits.

Ultimately, by integrating both distance-based and velocity-based scaling approaches, the calculation of object movement in SIT6 is determined by the following equation:

$$\Delta P_{object} = \Delta P_{touch} * (c * D_{object}) * min(\frac{Velocity_{touch}}{SC}, 1.2) \tag{3.1}$$

Where ΔP_{object} is the object translation distance, ΔP_{touch} is the touch translation distance, c is the fixed coefficient, D_{object} is the distance between the object and the user, $Velocity_{touch}$ is the current touch velocity, and SC is a predefined scaling constant.

The chosen values for the predetermined coefficient (c) were 0.001 for both translations in the XZ-plane and translations along the Y-axis. These values were selected empirically during implementation by considering the scale of the environment of the test application and the order of magnitude of the input readings from the touch surface. As for the scaling constant (SC), we decided on a value of 3000, corresponding to a 1:1 input mapping when the user's finger is moving at a velocity of 2.2cm per fixed update, a value also selected empirically. This fixed update interval is precisely 0.02s and corresponds to Unity's default time value for the *FixedUpdate* function.

3.2 Implementation

We used an HTC Vive Pro 2 HMD and a 32-inch infrared multi-touch frame for our solution's hardware, allowing interaction with the VR environment and enabling gesture input. Regarding software, we leveraged the Unity game engine as our development platform, which provided a robust framework to handle the input from the touch surface. From Unity's input system, we specifically employed its Touch API, which allowed us to retrieve the necessary information from the touch input to identify and enable our approach's gestures. The system architecture diagram presented in Figure 3.2 characterizes the organization of both software and hardware components.

3.2.1 Controller

At the core of our approach's controller component, we have employed a state machine to enable seamless transitions between the various gestures of the technique. This state-based implementation integrates error-handling mechanisms to ensure the proper functionality of gestures even in

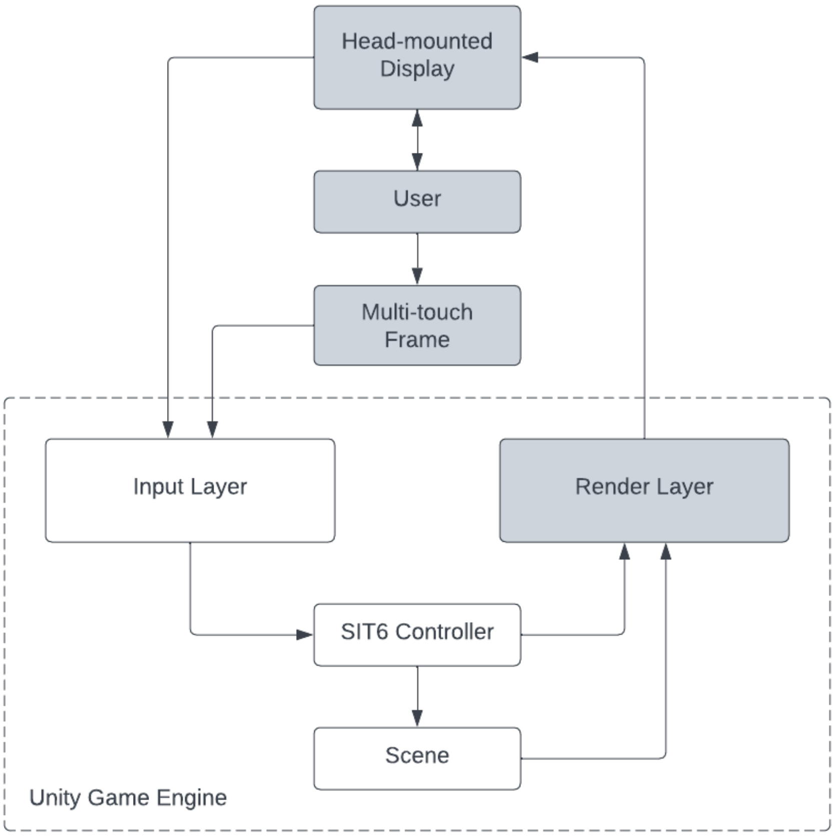


Figure 3.2: System architecture - the implemented components are depicted in white.

the presence of unexpected inputs. The state machine diagram displayed in Figure 3.3 illustrates a simplified version of how the controller operates, with Appendix B providing additional in-depth information with a flowchart diagram.

3.2.1.1 State Definitions

The state machine consists of several states, each representing a specific phase or gesture detected during user interaction. The following states are defined:

- **Idle**: This represents the controller's initial state, where it waits for user input. The controller checks for touch inputs in this state and transitions to the *Checking* state if a valid touch is detected.

- **Checking**: During this state, the controller conducts continuous checks on the touch inputs over a fixed period of time to determine the type of gesture the user performs. The system recognizes and classifies translation and rotation gestures by analyzing the touch positions and evaluating distances and angles between them. Based on this analysis, it then transitions to the corresponding gesture-specific states: *TranslationXZ*, *TranslationY*, *RotationX*, *RotationY*, or *RotationZ*.

- **TranslationXZ**: This state handles the translation gesture in the XZ-plane. It involves calculating the touch distance and applying distance and velocity-based scaling to get the object's new position. Then, it performs XZ translation accordingly, updating the object's position in the 3D environment based on the touch movement.

- **TranslationY**: Similarly to the *TranslationXZ* state, this state handles the translation gesture along the Y-axis. It calculates the touch distances and applies scaling based on distance and velocity to get the object's new position, performing Y translation accordingly.

- **RotationX**: This state handles rotation gestures around the X-axis. It calculates the vertical touch distance and then multiplies it by a fixed coefficient to obtain the rotation angle. Subsequently, it performs X-axis rotation on the object using the calculated angle.

- **RotationY**: This state manages the rotation gesture around the Y-axis. It calculates the angle between the vector formed by the previous two touch positions and the vector formed by the current two touch positions. Then, it utilizes this calculated angle to rotate around the Y-axis, updating the object's orientation accordingly.

- **RotationZ**: Similar to the *RotationX* state, this state handles rotation gestures around the Z-axis. It utilizes the horizontal touch distance and applies a fixed coefficient to it to calculate the rotation angle. It then performs the corresponding Z-axis rotation on the object with the computed angle.

3.2.1.2 State Transitions

In our implementation, a state check function is invoked at a fixed update interval of 0.02 seconds, Unity's default time value for the *FixedUpdate* function. This function plays a crucial role in our controller's state machine by evaluating various conditions to trigger state transitions. These conditions include changes in touch input, the detection of specific gestures, or the absence of touches. The state transitions are outlined as follows:

- From **Idle** to **Checking**: This change occurs when at least one touch is detected. During the transition, the first one or two touch positions are saved, serving as the reference points for subsequent calculations and gesture recognition at the end of the *Checking* state.

- From **Checking** to **TranslationXZ**, **TranslationY**, **RotationX**, **RotationY**, or **RotationZ**: This transition ensues when the state machine identifies a specific gesture based on the recorded touch positions and calculated distances and angles during the *Checking* state fixed analysis period which lasts, in our case, 0.06s (or three updates).

- From **Checking** back to **Idle**: This transition happens when the number of touches on the screen changes during the *Checking* state fixed analysis period, indicating a new touch has been added, or an existing touch has been removed.
- From **TranslationXZ**, **TranslationY**, **RotationX**, **RotationY**, or **RotationZ** back to **Idle**: This shift occurs when the current touch positions no longer indicate valid motion for that gesture state or when all touches are released.

3.2.1.3 Gesture Detection

During the analysis time of the *Checking* state, several conditions and thresholds were defined for activating each state in the state check function. These conditions are as follows, with their corresponding fixed threshold values being noted posteriorly:

- **TranslationXZ**: To activate this state, the user must have only one active touch and demonstrate touch movement in any direction that exceeds the required minimum threshold.
- **TranslationY**: The user must have two active touches to trigger this state. One of the touches should remain stationary below the maximum threshold of movement. In contrast, the other touch, the moving finger, must exceed the minimum required threshold of vertical movement without surpassing the maximum limit of horizontal movement.
- **RotationX**: To activate this state, the user must have two touches. Both touches need to exceed the minimum required threshold of vertical movement while staying within the maximum limit of horizontal movement.
- **RotationY**: Two active touches from the user are required to trigger this state. The angle between the vector formed by the previous two touch positions at the time of the transition between the *Idle* state and the *Checking* state and the vector formed by the current two touch positions at the end of the *Checking* state analysis period must exceed the minimum angle required for the motion to be recognized as a rotation. Additionally, both fingers' horizontal and vertical movements must surpass a minimum threshold.
- **RotationZ**: In order to activate this state, the user must have two touches. Similar to the activation condition for the *RotationX* state, both touches must surpass the minimum required threshold of horizontal movement while remaining within the maximum limit of vertical movement.

Every fixed limit and threshold has been meticulously selected to ensure user gestures consistently trigger the intended actions. Thus, by taking into account the size of the multi-touch frame, the following values for the *Checking* state analysis period (0.06s) were selected:

- **Minimum activation movement (All states)**: 25px $\approx$ 1cm;
- **Maximum stationary movement (TranslationY)**: 5px $\approx$ 0.2cm;

- **Maximum horizontal movement (TranslationY / RotationX)**: 50px $\approx$ 2cm;
- **Maximum vertical movement (RotationZ)**: 50px $\approx$ 2cm;
- **Minimum activation angle (RotationY)**: 10°.

Moreover, the order in which the conditions for each state are checked has been carefully arranged by considering the difficulty level of the gestures required to activate each state. Higher difficulty gestures, such as those for *RotationY* and *TranslationY*, have been given priority in the sequence. The complete state-checking order for two-finger gestures is the following: *TranslationY* $\rightarrow$ *RotationY* $\rightarrow$ *RotationZ* $\rightarrow$ *RotationX*.

3.2.1.4 Error Handling

A vital component of the error-handling mechanisms in our solution involves using a threshold variable to monitor consecutive updates (occurring every 0.02s) of touch input that fail to satisfy the conditions for a specific gesture state. This threshold variable serves as a countdown mechanism, decrementing each frame when the conditions for the current state are not met. When the threshold variable reaches zero, the touch input significantly deviates from the expected motion, necessitating a transition back to the *Idle* state. This feature guarantees that the system does not abruptly switch states when a valid touch is temporarily not detected.

Furthermore, we have implemented a mode-locking mechanism for gesture states that require two touches. Upon entering one of these states (*TranslationY*, *RotationX*, *RotationY*, and *RotationZ*), the system remains locked in that state even if the user introduces additional touches. The two original touches retain control over the transformations, disregarding any subsequent touch inputs. The state can only be changed when the total touch count drops below two. In this case, as the threshold variable reaches zero, indicating a sustained deviation from the expected touch input, the controller transitions to the *Idle* state. This state-locking prevents the transformations from stopping if the user inadvertently adds more touches.

Hence, incorporating these error-handling mechanisms ensures a seamless and fluid user interaction experience, effectively addressing any potential movement errors. These mechanisms are particularly crucial due to the inherent limitations of the multi-touch frame, which exhibits both high sensitivity and limited detection precision. As a result, accidental touches by users and frequent loss of valid touches are common occurrences that must be managed efficiently. Along with these error-handling mechanisms, the implemented visual indicators explained in Subsection 3.2.2 also help mitigate some of the previously mentioned problems.

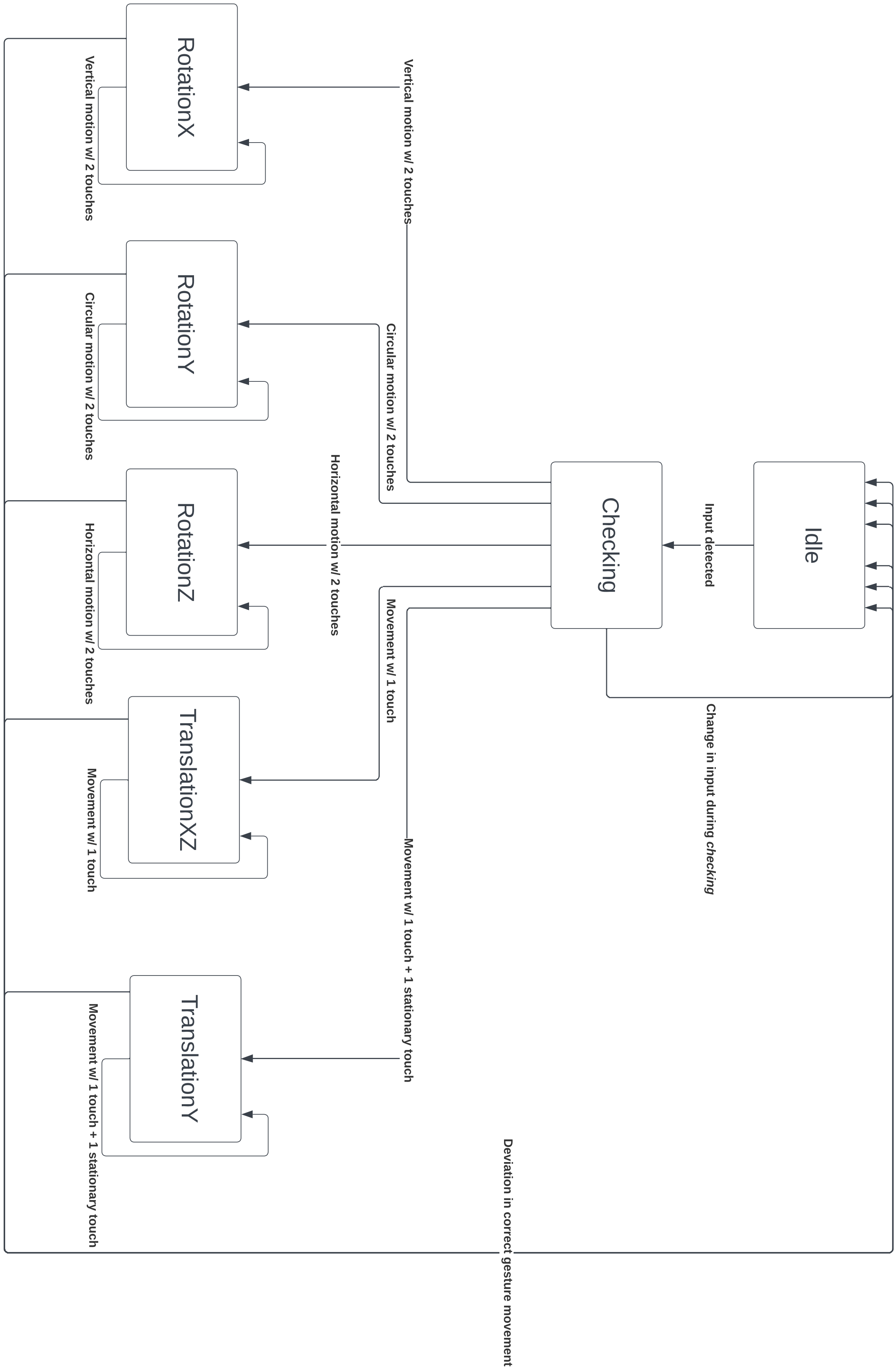


Figure 3.3: Simplified state-machine diagram of SIT6’s controller .

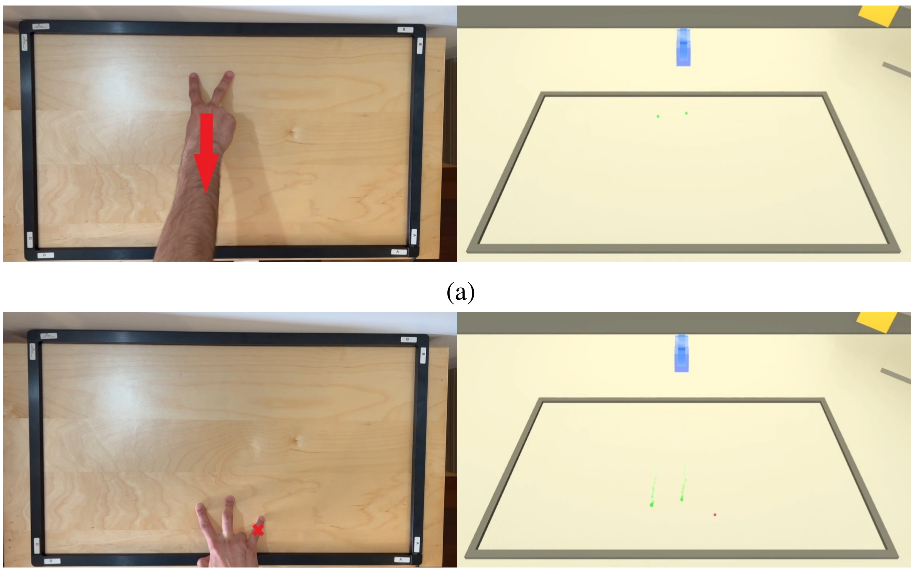
3.2.2 Visual Indicators

To aid users in performing and distinguishing gestures within the VR environment, our implementation of SIT6 incorporates two distinct visual indicators. These indicators provide clear visual cues to enhance user gesture recognition and comprehension.

3.2.2.1 Virtual Touch Frame

The first indicator tool utilized in our system consists of a virtual representation of a multi-touch frame. This virtual model accurately replicates the dimensions of the real-life multi-touch frame, ensuring a faithful representation within the virtual environment. Whenever the user makes contact with the touch surface, the virtual representation displays the precise position of their fingers. The touches actively controlling the transformations are visually highlighted in green, representing valid touches. Conversely, touches with no control function, typically any contact beyond the initial two touches, are depicted in red. The virtual multi-touch frame model and its functionality are shown in Figure 3.4.

This visual mechanism enables users to consistently track the placement of their fingers, addressing the inherent challenge of not being able to see their hands physically in virtual reality. By providing continuous feedback on hand position, our system allows users to verify whether they have made any accidental touches and determine if the executed gesture corresponds to their intended action. This feature facilitates the identification and correction of any mistakes made during the gesture execution process, which is particularly valuable given the high sensitivity and limited precision of the multi-touch frame utilized.



(a)

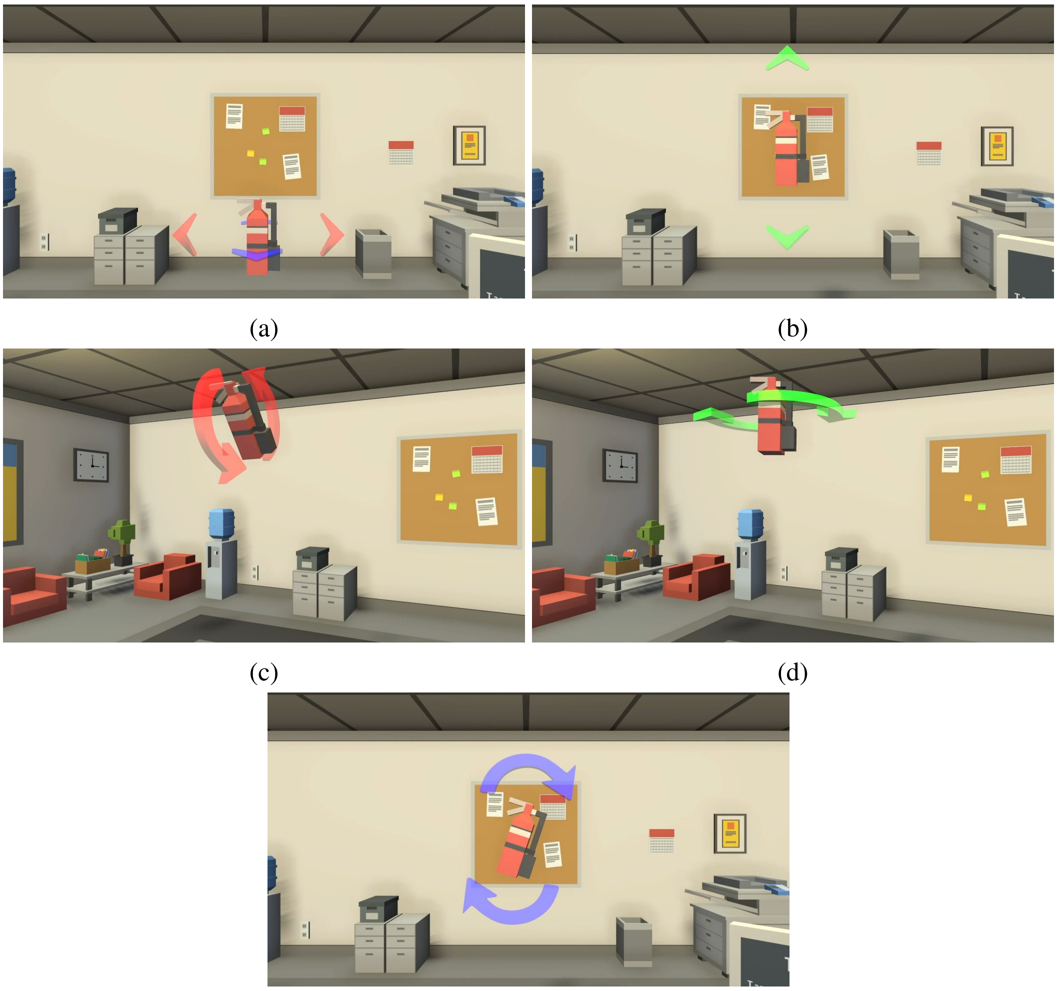
(b)

Figure 3.4: Virtual touch frame indicator in action. (a) The two green points on the virtual touch frame (right) represent the position of the fingers (left). (b) The invalid third touch, marked with a cross, is represented in red on the virtual frame.

3.2.2.2 Axis Indicator Arrows

In addition, we incorporated an indicator tool that utilizes arrows to represent the axis along which the object is undergoing translation or rotation, illustrating the direction and orientation of the object's movement. The shape of the arrows dynamically changes based on whether the user is performing a rotation or a translation, while their color depends on the specific axis on which the operation is being executed: red for the X-axis, green for the Y-axis, and blue for the Z-axis. Figure 3.5 depicts the different arrow indicators.

The implementation of these arrow indicators provides users with a visual aid that enables a clearer understanding of the specific axis involved in the transformation. This tool is especially beneficial during rotations, which can be challenging to interpret in three-dimensional space, as the axis of rotation and subsequent movement may not correspond with the user's expectations. As a result, the ability to control and manipulate objects in the virtual environment improves, significantly enhancing our solution's overall usability and effectiveness.



(a) (b) (c) (d) (e)

Figure 3.5: Axis indicator arrows: (a) XZ-plane translation. (b) Y-axis translation. (c) X-axis rotation. (d) Y-axis rotation. (e) Z-axis rotation.

3.3 Summary

In this chapter, we introduced a novel object manipulation technique called Scaled Indirect Touch 6-DOF, or SIT6, tailored for DeskVR environments and designed to address precision and out-of-reach interactions. Section 3.1 introduces the design of our proposed solution, which integrates a gesture dictionary derived from the Indirect6 technique. The modifications to the original design aimed to address the distinctive requirements of virtual reality applications and included adjusting the translation gestures to align with the XZ-plane instead of the XY-plane and swapping the rotation gestures for yaw (Ry) and roll (Rz). Furthermore, the proposed solution utilizes a distance and velocity-based mapping approach based on Scaled HOMER. For distance-based scaling, our solution uses the distance between the user and the object multiplied by a coefficient. As for the velocity-based scaling, it employs the finger velocity divided by a scaling constant. The calculation of object movement in SIT6 is determined by a formula that incorporates both scaling approaches.

Section 3.2 detailed our solution's implementation and presented the hardware and software components required for it. The hardware setup included an HTC Vive Pro 2 HMD and a 32-inch infrared multi-touch frame. As for software, we relied on the Unity game engine as the development platform. The controller component of the solution is implemented as a state machine that seamlessly transitions between different gestures based on touch input. Furthermore, the controller incorporates error-handling mechanisms to ensure a smooth user experience, including a threshold variable that only switches states when multiple consecutive updates deviate from expected motion and a mode-locking mechanism that prevents blocking for two-finger gestures. In addition to the controller component, our implementation of SIT6 incorporates two visual indicators to enhance user gesture recognition and comprehension in the VR environment: a virtual touch frame model that precisely displays finger positions and axis indicator arrows that show object movement direction, dynamically changing shape and color based on the performed transformation and employed axis.

Chapter 4

User Evaluation

A series of experiments were conducted to assess our solution's effectiveness and answer the research questions, consequently validating or refuting the proposed hypothesis of this work. Participants were assigned docking tasks to perform, allowing us to collect the necessary quantitative and qualitative data for evaluating the proposed method's performance and usability. These docking missions were designed to emulate real-life professional tasks in contexts such as construction or architecture, scenarios requiring both precise and out-of-reach manipulation, along with complex structure visualization.

To facilitate a comprehensive comparison, we implemented an alternative method already analyzed in previous articles: Scaled HOMER [38]. This approach makes for a reliable baseline since it is a functional and already well-established object manipulation technique that shares some characteristics with SIT6.

4.1 Baseline

The comparison baseline should be composed of techniques that, although not specifically designed for a DeskVR scenario, have characteristics that qualify them to be implemented in this context, allowing us to test our solution's performance against them. Therefore, based on the Discussion section of the Related Work chapter (2.3), the only technique with the necessary features to be included in this baseline is Scaled HOMER. Although mid-air, this approach attempts to tackle the same problems as SIT6, making it a solid ground comparison in terms of effectiveness.

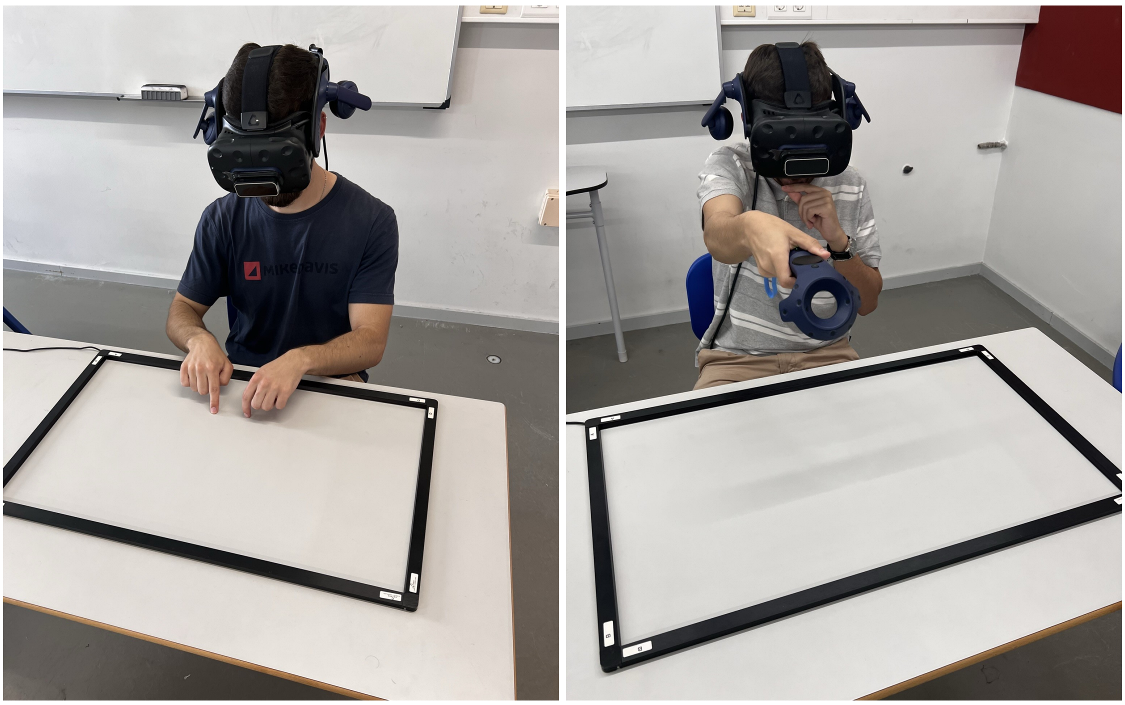
This technique was implemented using the Unity game engine and the SteamVR plugin, which contains valuable programming tools that allowed us to work with the VR controllers necessary for the technique to function. While Scaled HOMER was designed to work in VR environments and was employed for object manipulation (through a virtual hand), it was also meant to be used in object selection (through raycasting). Therefore, we had to adapt the method to restrict it to work only for manipulation in order to make it an acceptable comparison method for SIT6.

With this in mind, we made it so that when the user presses and holds the controller's trigger button, the object is selected and the manipulation process starts immediately. The technique then works as expected, with the object following the user's hand as long the trigger is held. This mechanism allows the user to reset the position of their hands when their arm's length is reached by letting go of the trigger and then readjusting their hand's position before pressing the trigger again, without the need to select the object. Thus, this change removes the need for raycasting selection entirely.

4.2 Setup

We employed the HTC Vive Pro 2 virtual reality HMD, the complementary controllers, and a 32-inch infrared multi-touch frame for the experiment's hardware. The HMD and touch surface were connected to a VR-compatible laptop equipped with an AMD Ryzen 7 5800H CPU and an Nvidia RTX 3070 GPU. The computer had all the necessary software and drivers installed.

Throughout all experiments, participants were seated at a desk with the touch frame and VR controller on top, as seen in Figure 4.1. Considering the high sensitivity of these devices, we deactivated them when not in use to prevent interference with the testing software and methods.

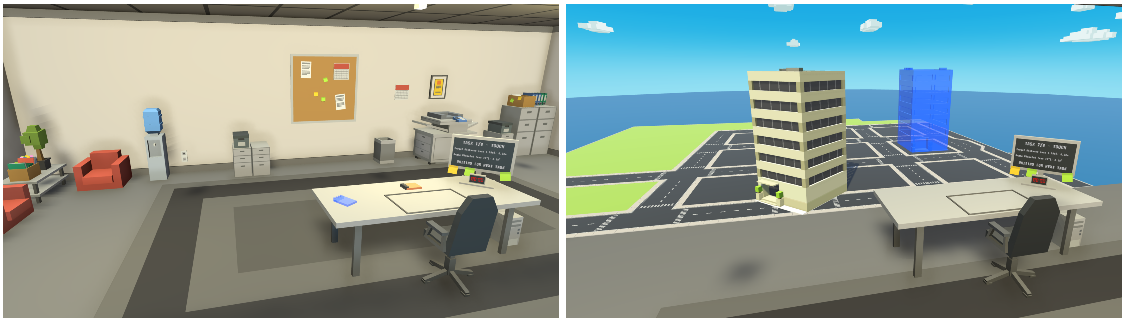


(a) (b)

Figure 4.1: Some participants of the experiments. (a) Setup for the SIT6 tasks. (b) Setup for the Scaled HOMER tasks.

4.3 Test Environment

The test environment for the user study consisted of an office setting, which would later expand to include the outside city streets as the experiment progressed, as depicted in Figure 4.2. The scene was scaled and adjusted to precisely match the touch frame's physical position and size. Within this environment, participants were seated at a desk with a timer and computer monitor. The monitor displayed vital information, including details about the current task, the docking status, and any position and rotation mismatches between the object and the docking point.



(a) (b)

Figure 4.2: Settings for the test environment: (a) Office interior for the first four tasks. (b) City streets for the last four tasks.

4.4 Methodology

The experimental procedure for the user study consisted of three phases, carried out sequentially. This procedure was repeated once for each tested method, alternating the tested technique order among participants.

Before initiating the experiments, participants were asked for consent to share the images and test data collected during the procedure. Additionally, they were requested to complete a profiling questionnaire, which assisted in better understanding their background and characteristics for the study. After that, participants received a brief video presentation that provided an overview of each object manipulation technique and outlined the objectives of the tasks they would be undertaking.

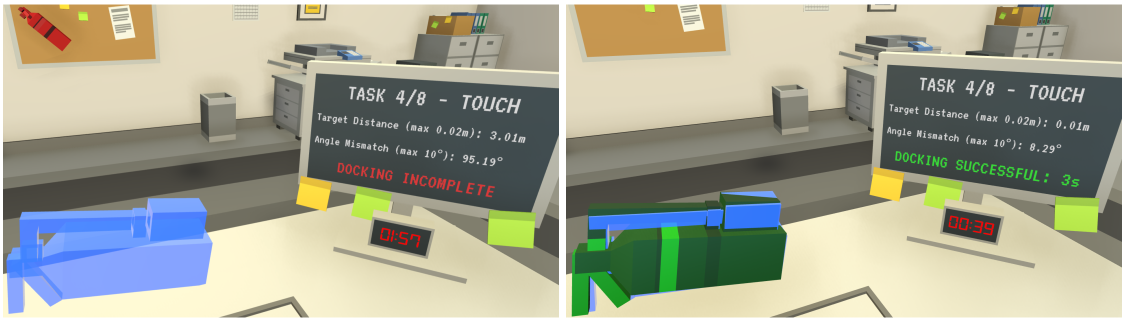
The first phase allowed participants five minutes to practice the input mappings until they became comfortable with the experimented method. This practice phase contained four docking tasks that could be completed repeatedly. Since this step was considered training, the data collected in this phase was not part of the analysis.

The second stage comprised a sequence of timed trial docking tasks, with each task having a time limit of 2 minutes. This time boundary was necessary to keep the duration of the study manageable, with the ongoing task concluding if the limit was reached, subsequently progressing to the next task. Once all the tasks were completed, the third phase focused on gathering participant feedback regarding their interaction experience with the tested technique. This feedback was obtained through a questionnaire. A comprehensive explanation of the tasks is provided in section 4.5.

4.5 Tasks

Each task consisted of a 6-DOF 3D docking mission parallel to the one used in the study for Indirect6 [31]. This docking task involved moving an object from one position to another. A transparent blue object visually represented the desired final position for the manipulable object, referred to as the docking point. We established a threshold for evaluating whether or not the object was in its final position, particularly to facilitate placements involving distant objects: the distance between the docking point and the object should be within 2% of the distance between the docking point and the camera, and the angle mismatch between the two positions should not exceed 10 degrees.

When the manipulable object fell within the specified threshold of the docking point, its color transitioned to green, and the task would be deemed successful after a 5-second countdown, provided the object did not leave its position. Conversely, while the placement was incomplete, the object would blink red. Notably, the trial timer only started after the experiment's moderator pressed the start button, granting precision to the experiment results and avoiding accidental starts. Figure 4.3 illustrates the docking task procedure.



(a) (b)

Figure 4.3: The docking task process. (a) Incomplete docking task: the object (fire extinguisher) blinks red. (b) Complete docking task: the object's color changes to green.

The experiment encompassed a total of eight docking tasks, each featuring a distinct combination of sizes, positions, and rotations for both the manipulable object and the docking point. These varied configurations were designed to assess the effectiveness of the approaches in scenarios involving out-of-reach and precision manipulations. The eight tasks are categorized into four classes based on the distance between the user and the docking point or the user and the initial object position. Additionally, the object size increases proportionally with distance in each class. These classes are defined as follows:

- **Ultra-Close**: object or docking point is on the desk (0 m - 1.5 m);

- **Close**: object or docking point is inside the office (1.5 m - 5 m);
- **Medium**: object or docking point is between the office and the city (5 m - 55 m);
- **Far**: object or docking point is in the city or beyond (>55 m).

Within each class, there are two tasks, which comprise different difficulty levels. While both tasks require translation movement, either in 2 DOFs (XZ-plane) or 3 DOFs, the first task is more straightforward and solely requires rotation along a single axis. In contrast, the second task entails a more intricate rotation involving multiple axes.

To maintain consistency among participants, the order of the tasks remained fixed and identical throughout every experimental procedure. The predetermined task sequence started with *Ultra-Close* distance tasks involving objects placed on the desk, which was followed by *Close tasks* with objects positioned within the office environment. Subsequently, the test environment transitioned, revealing the outside cityscape, enabling the undertaking of *Medium* tasks and later *Far* tasks involving objects situated outside the office. The specific order for the task sequence is the following:

Ultra-Close Simple → *Ultra-Close Complex* → *Close Simple* → *Close Complex* → *Medium Simple* → *Medium Complex* → *Far Simple* → *Far Complex*

The details involving each task, which include object size, distances, rotations (in Euler angles), and the necessary transformations, are the following:

Ultra-Close Simple (Notepad) - Figure 4.4

- **Object size**: (0.13, 0.04, 0.18) m;
- **Distance between the user and the object**: 0.94 m;
- **Distance between the user and the docking point**: 1.13 m;
- **Distance between the object and the docking point**: 0.82 m;
- **Rotation between the docking point and the object**: (0.00, 60.00, 0.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Rotation in the Y axis.

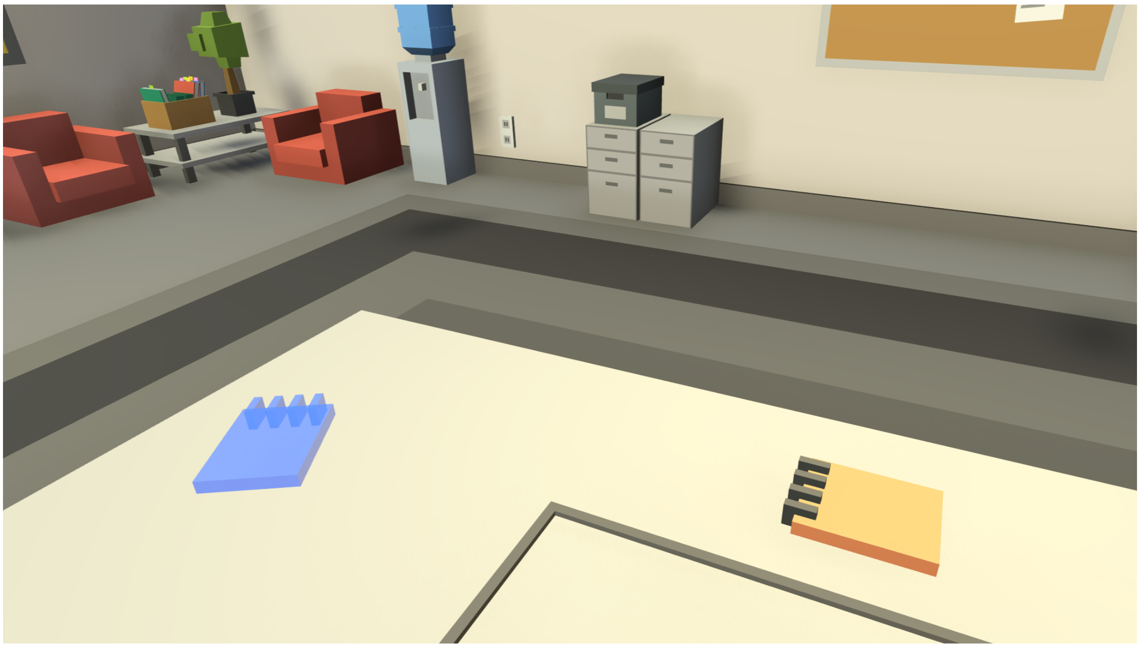


Figure 4.4: Ultra-Close Simple (Notepad) task.

Ultra-Close Complex (Photograph) - Figure 4.5

- **Object size**: (0.13, 0.13, 0.07) m;
- **Distance between the user and the object**: 1.11 m;
- **Distance between the user and the docking point**: 1.00 m;
- **Distance between the object and the docking point**: 1.21 m;
- **Rotation between the docking point and the object**: (-83.00, -250.00, -180.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in 2-3 axes.

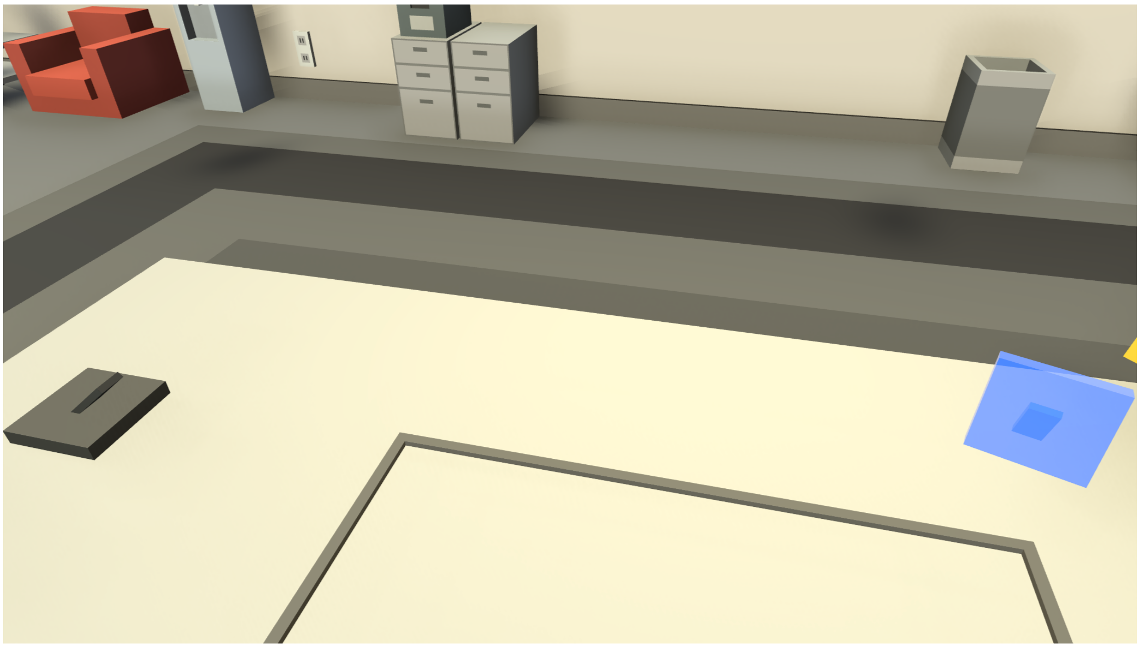


Figure 4.5: Ultra-Close Complex (Photograph) task.

Close Simple (Calendar) - Figure 4.6

- **Object size**: (0.30, 0.29, 0.01) m;
- **Distance between the user and the object**: 1.02 m;
- **Distance between the user and the docking point**: 3.86 m;
- **Distance between the object and the docking point**: 3.20 m;
- **Rotation between the docking point and the object**: (90.00, 00.00, 00.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in the X axis.

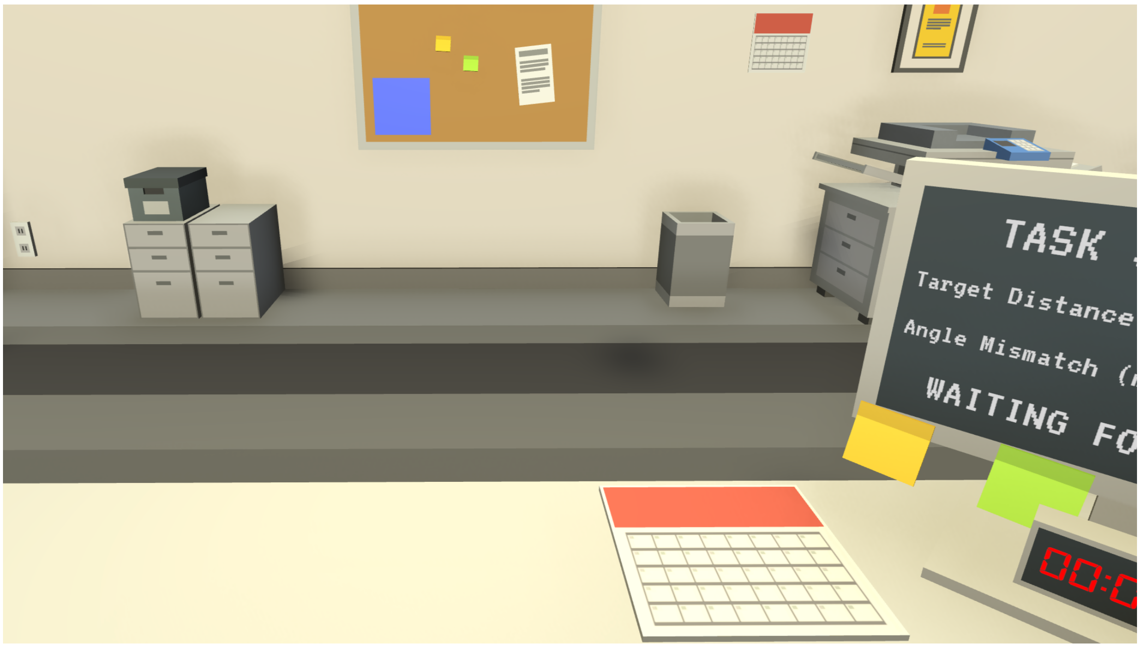


Figure 4.6: Close Simple (Calendar) task.

Close Complex (Fire Extinguisher) - Figure 4.7

- **Object size**: (0.26, 0.43, 0.12) m;
- **Distance between the user and the object**: 3.73 m;
- **Distance between the user and the docking point**: 0.92 m;
- **Distance between the object and the docking point**: 3.01 m;
- **Rotation between the docking point and the object**: (-45.00, 90.00, 80.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in 2-3 axes.

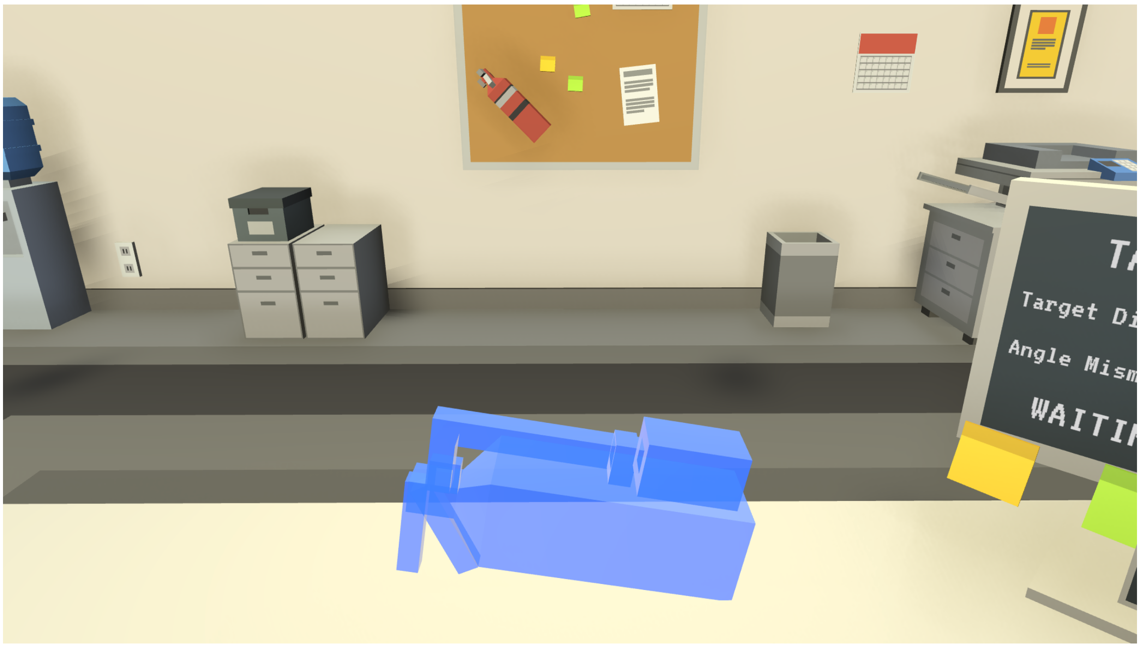


Figure 4.7: Close Complex (Fire Extinguisher) task.

Medium Simple (Hot Air Balloon) - Figure 4.8

- **Object size**: (5.36, 8.61, 5.43) m;
- **Distance between the user and the object**: 9.61 m;
- **Distance between the user and the docking point**: 29.68 m;
- **Distance between the object and the docking point**: 24.42 m;
- **Rotation between the docking point and the object**: (00.00, 00.00, -60.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in the Z axis.

Figure 4.8: Medium Simple (Hot Air Balloon) task.

Medium Complex (Airplane) - Figure 4.9

- **Object size**: (7.93, 2.37, 10.25) m;
- **Distance between the user and the object**: 27.58 m;
- **Distance between the user and the docking point**: 52.74 m;
- **Distance between the object and the docking point**: 37.42 m;
- **Rotation between the docking point and the object**: (-45.00, -180.00, 45.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in 2-3 axes.

Figure 4.9: Medium Complex (Airplane) task.

Far Simple (Building) - Figure 4.10

- **Object size**: (10.90, 21.79, 12.13) m;
- **Distance between the user and the object**: 53.92 m;
- **Distance between the user and the docking point**: 90.59 m;
- **Distance between the object and the docking point**: 46.40 m;
- **Rotation between the docking point and the object**: (00.00, -45.00, 00.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Rotation in the Y axis.

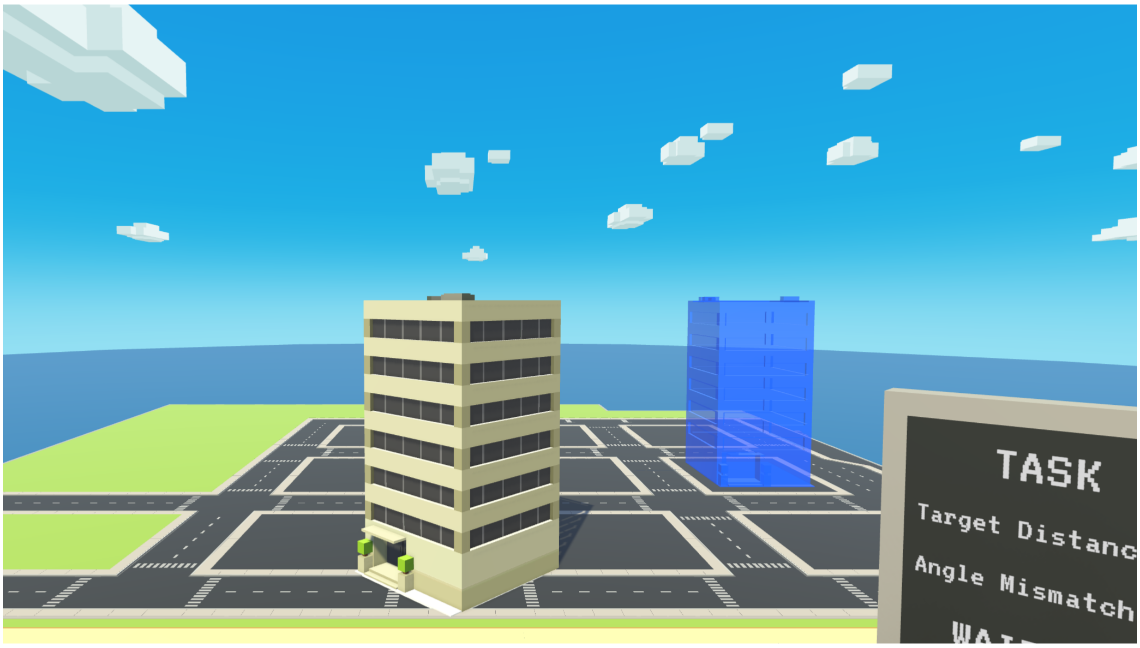


Figure 4.10: Far Simple (Building) task.

Far Complex (Shopping Mall) - Figure 4.11

- **Object size**: (22.80, 5.08, 25.91) m;
- **Distance between the user and the object**: 90.60 m;
- **Distance between the user and the docking point**: 126.37 m;
- **Distance between the object and the docking point**: 82.61 m;
- **Rotation between the docking point and the object**: (-90.00, -150.00, -180.00)°;
- **Necessary transformations for completion**: Translation in the XZ plane, Translation in the Y axis, Rotation in 2-3 axes.

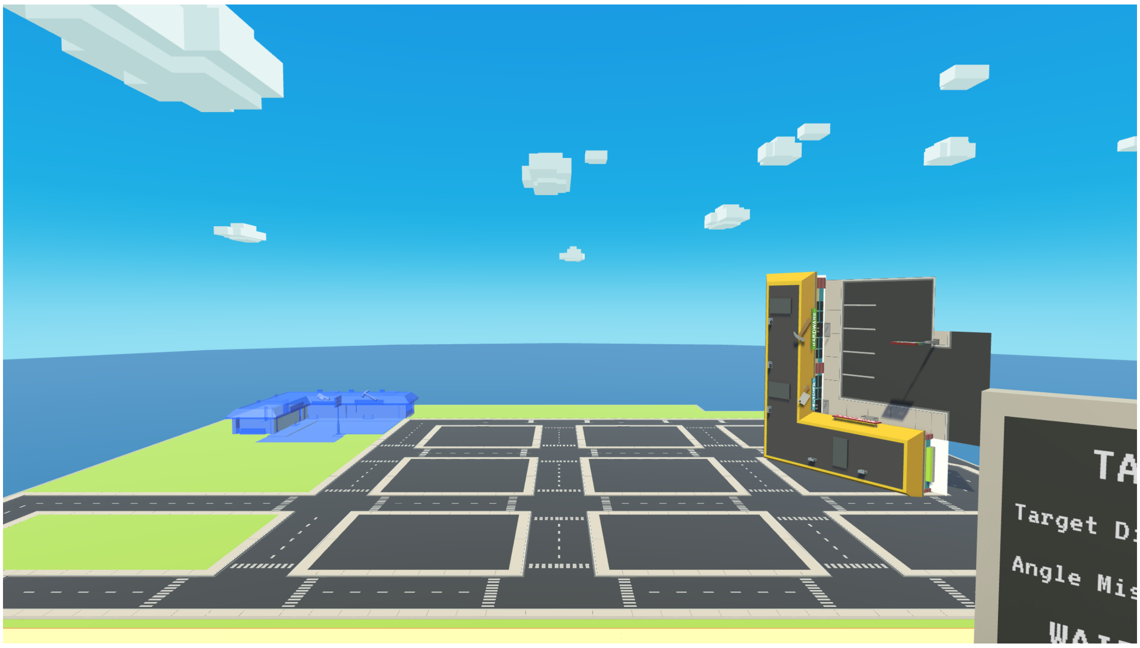


Figure 4.11: Far Complex (Shopping Mall) task.

4.6 Data Logging

In order to define the analyzed metrics to assess the performance differences between the techniques following the experiments, we gathered quantitative and qualitative data from the users.

4.6.1 Objective Data

Regarding the objective data, we logged each task's success rate, calculated by dividing the number of successful completions of a specific task with a technique by the total number of participants (total task attempts with a technique). Furthermore, we recorded the completion times (in seconds) of the successful tries, designating the duration required to move the object from the starting position to the goal position. At the end of each task, we also saved data regarding the transformations applied to the object. This information includes the time spent on each type of transformation, the total distance translated, and the total angle rotated for each technique. Moreover, we also logged the object's idle time.

For redundancy, we meticulously logged additional information frame by frame, encompassing the object's position and rotation, as well as the distance and rotation mismatch between the object and the docking point. For SIT6, we specifically logged the touch count and the positions of the two valid touches, along with the current state of the interaction. In the case of Scaled HOMER, we recorded the position and rotation data of the VR controller, along with the status of whether or not the user was holding the trigger. All objective data was measured and logged automatically by the system.

4.6.2 Subjective Data

As for the subjective data, we aimed to record user feedback on comfort level, perceived demand, ease of use, ease of learning, and DeskVR suitability for each method. Since this data is subjective and varies among users, we gave participants a questionnaire containing six statements. In this form, users had to rate their agreement with each statement according to a 5-point Likert Scale, ranging from 1 ("Strongly Disagree") to 5 ("Strongly Agree"), selecting the option that best described their interaction experience with the tested method. For more details on the statements of this questionnaire, refer to Appendix A.

4.7 Participants

A total of 26 individuals participated in the experiments, consisting of 20 males and 6 females, all right-handed. All participants consented to participate in the experiment, share their test data, and allow the capture of any images during the procedure. Among the participants, 23 were aged between 21 and 30, with 1 participant falling in the 16 to 20 age range and 2 participants being over 40 years old.

Regarding academic qualifications, 20 participants held a bachelor's degree, 1 had a master's degree, 2 had a high school degree, and 3 had completed middle school. Among the participants with a degree or currently enrolled in university, 17 studied computer engineering, 5 pursued other engineering fields, and 1 studied law.

In terms of employment, 22 participants were students, 2 worked in engineering, and 2 had other types of jobs. Regarding the participants' prior experience with virtual reality, 14 had never used VR, 10 had used it once or twice, 1 had an annual usage, and only 1 participant used VR daily.

4.8 Summary

In this chapter, we described the series of experiments conducted to evaluate the effectiveness of our proposed approach and answer the research questions. Section 4.1 established and detailed our implementation of the baseline of comparison: a modified version of Scaled HOMER focusing solely on object manipulation by removing the object selection process. The experimental setup, described in section 4.2, included the HTC Vive Pro 2 virtual reality headset, the complementary VR controllers, and a 32-inch infrared multi-touch frame. Moreover, section 4.3 outlines the test environment for the user study, which consisted of an office setting that dynamically expanded as the experiment progressed.

Section 4.4 defined the experimental procedure, which comprised three phases: technique practice, timed trial docking tasks, and participant feedback. In the practice phase, participants had 5 minutes to familiarize themselves with the tested method. Next, in the timed trial docking tasks sequence, participants were given 2 minutes to complete each task, which involved moving objects from one position to another. After completing the task sequence, participants provided feedback through a questionnaire. Section 4.5 provided further details on the tasks, with the sequence comprising eight tasks grouped into four classes based on the distance between the user and the object or docking point. The tasks varied in difficulty based on the required transformations.

Section 4.6 presented the quantitative and qualitative data that was collected from users during the experiments. Regarding objective data, these included the tasks' success rate, completion times, and transformation details. Additional frame-by-frame data was also logged for both techniques, capturing object/controller/touch positions and rotations. As for subjective data, we focused on gathering information on user experience. Participants rated their agreement with six statements using a 5-point Likert Scale, indicating their personal experience with the method. The statements covered comfort, physical and mental demand, ease of use, ease of learning, and suitability for the task.

Lastly, section 4.7 provided information about the participants involved in the experiments conducted. It included several details, including the total number of participants, the distribution of age and gender, academic qualifications, fields of study, and employment situations. Furthermore, the section also mentioned the participants' dominant hand and prior experience with virtual reality.

Chapter 5

Results

In this chapter, we present, analyze and discuss the results obtained from the conducted user experiments outlined in chapter 4. The focus of this section is to present a comprehensive analysis of the quantitative and qualitative data collected during the experimentation process.

By examining objective metrics such as completion time, idle time, active time, total object translation, and total hand movement, we aim to evaluate the technique's effectiveness and efficiency in executing tasks. Similarly, we also analyze subjective metrics, which include user feedback on comfort, perceived physical and mental demand, ease of use, ease of learning, and DeskVR context suitability. By exploring these subjective aspects, we stand to gain valuable insights into the user experience and their overall satisfaction with the proposed solution.

5.1 Metrics

As mentioned earlier, our data collection process involved gathering objective data after each individual task and subjective data at the end of a task sequence, allowing us to define several metrics to analyze for the results. For the qualitative metrics used in our analysis, we derived them directly from each question's response data provided in the user experiences form.

In contrast, when defining the quantitative metrics, we applied a filtering process to narrow down our collected data to the most informative subset for the study. Thus, our metrics for the analysis focused on task success rate, completion time, idle time, active time, total object translation, and total hand movement. Although, as previously mentioned in section 4.6, the system automatically logged most data for these metrics, the hand movement metric was obtained by posteriorly processing the collected frame-by-frame controller and touch positions.

5.2 Analysis

Several statistical tests were conducted on the defined qualitative and quantitative metrics, which were employed to draw conclusions from the gathered information. On these tests, we utilized the conventional alpha-value of 5% ($\alpha = 0.05$) to determine whether or not the results had a statistically significant difference.

5.2.1 Objective Results

To ensure the reliability of our quantitative metrics, we first identified and removed any outliers through descriptive data analysis. Subsequently, we applied the Shapiro-Wilk [29] test to determine whether the remaining data adhered to a normal distribution. Since we wanted to compare the result of two distinct techniques, we then employed either the paired-samples t-test [28] when the data followed a normal distribution or the Wilcoxon signed-rank test [27] when the data did not exhibit normality.

5.2.1.1 Completion Time

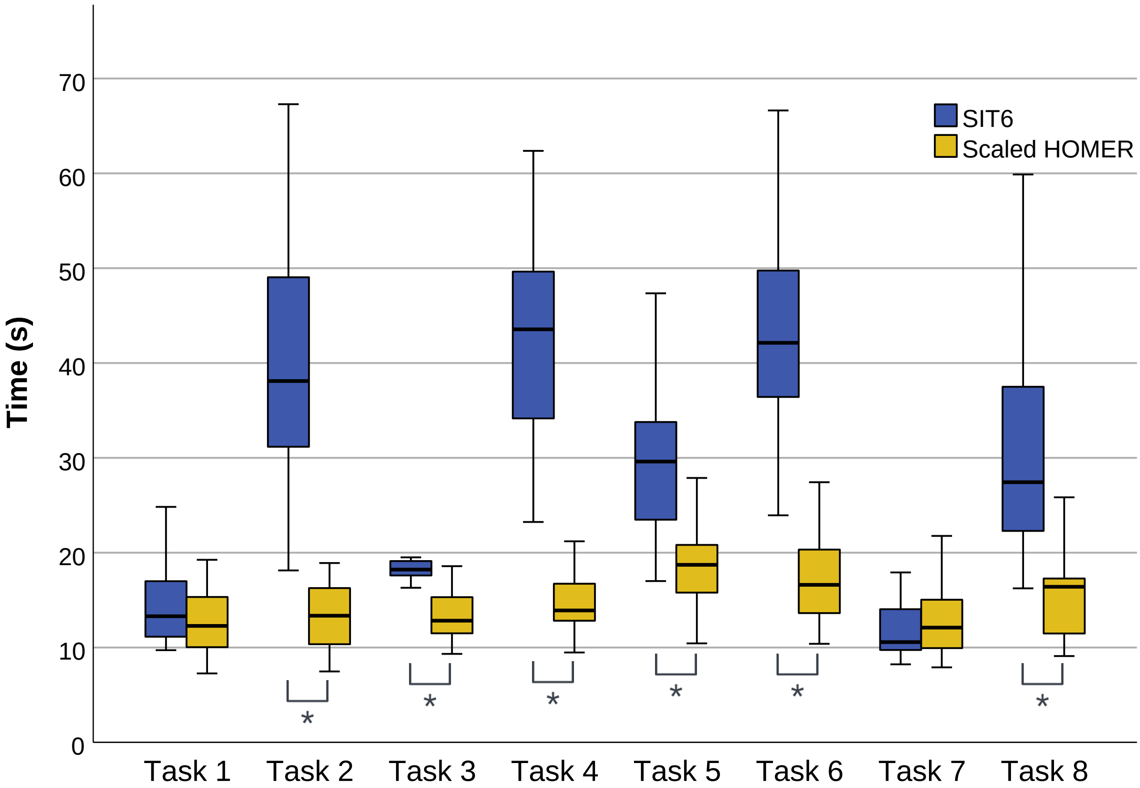


Figure 5.1: Box-plot of completion time per task by technique. * indicates statistically significant differences between the methods.

Since every participant achieved a task success rate of 100% for both techniques, we were able to measure the completion time for every task and technique across all users. This metric, recorded in seconds, is displayed in Figure 5.1 and provides the following results:

- **Task 1 (Ultra-Close Simple)**: No statistically significant difference in completion time between the two methods ($Z = -0.049, p = 0.961$).
- **Task 2 (Ultra-Close Complex)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(22) = 10.128, p < 0.001$), with $13.01 \pm 3.29s$ compared to $40.25 \pm 11.78s$.
- **Task 3 (Close Simple)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(10) = 6.459, p < 0.001$), with $12.98 \pm 2.57s$ compared to $18.36 \pm 0.97s$.
- **Task 4 (Close Complex)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(17) = 11.025, p < 0.001$), with $14.69 \pm 3.25s$ compared to $39.44 \pm 8.48s$.
- **Task 5 (Medium Simple)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(22) = 4.638, p < 0.001$), with $18.45 \pm 4.65s$ compared to $29.28 \pm 8.71s$.
- **Task 6 (Medium Complex)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(23) = 11.368, p < 0.001$), with $17.35 \pm 4.99s$ compared to $42.39 \pm 11.45s$.

- **Task 7 (Far Simple)**: No statistically significant difference in completion time between the two methods ($Z = -1.195, p = 0.232$).
- **Task 8 (Far Complex)**: Scaled HOMER was statistically significantly faster than SIT6 ($t(20) = 5.161, p < 0.001$), with $15.81 \pm 5.00s$ compared to $31.05 \pm 11.98s$.

Interpretation

After analyzing the completion time results, it is evident that Scaled HOMER significantly outperformed SIT6 in most tasks. This performance difference is especially noteworthy in complex tasks involving rotations across multiple axes (tasks 2, 4, 6, 8). The superior performance of Scaled HOMER is probably caused by its inherently natural manipulation, as the object follows the user's hand movements. Additionally, Scaled HOMER's approach of not separating degrees of freedom (DOFs) allowed participants to perform simultaneous rotations along multiple axes, resulting in faster complex task completion.

Despite anticipating similar performance between the two techniques in basic tasks 3 and 5, we observed a decline in performance with SIT6 compared to Scaled HOMER. The task completion time in SIT6 was possibly influenced by several factors. One of these factors was the high sensitivity of the infrared multi-touch frame, frequently leading participants to unintentionally trigger a rotation, which unfortunately typically occurred along an incorrect axis. Additionally, the state detection controller occasionally misinterpreted *RotationX* and *RotationZ* gestures as *RotationY* gestures when users did not place their fingers completely correctly. As a result, these combined factors transformed what were intended to be straightforward tasks into more complex ones that required rotation along multiple axes.

Nevertheless, in the case of simple tasks 1 and 7, where the objective was to translate and rotate the object solely along the Y-axis, an intriguing result emerged: both techniques showed comparable performance. This outcome is likely due to the challenging nature of executing Y-axis rotation gestures in Scaled HOMER, which frequently placed considerable strain on the user's wrist, thus slowing down the rotation movement. This issue was consistently observed among participants, with one individual, in particular, experiencing difficulty in Scaled HOMER tasks due to carpal tunnel syndrome (CTS). On the other hand, in SIT6, the RotationY state was prioritized over other rotation states in the state verification order, making the state easier to trigger, despite its gesture being more challenging to perform. This prioritization facilitated engagement with the *RotationY* state and reduced completion time for tasks requiring Y-axis rotation.

5.2.1.2 Idle Time

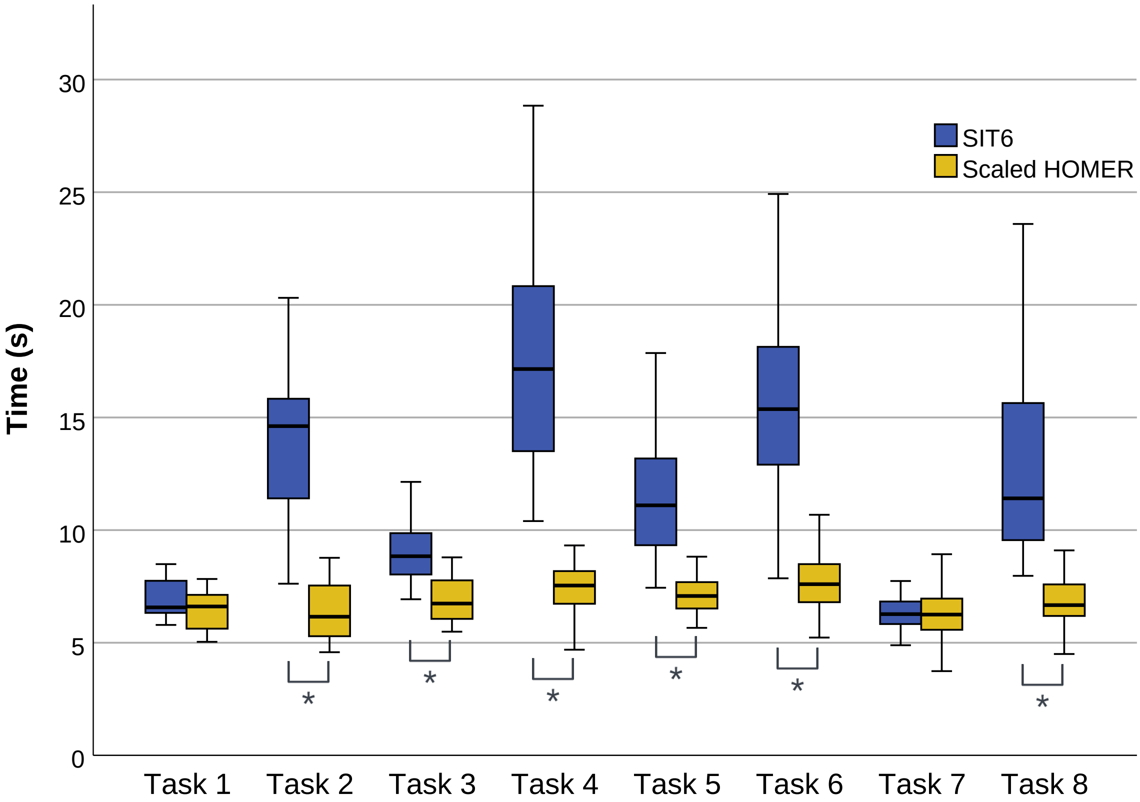


Figure 5.2: Box-plot of idle time per task by technique.

For the idle time metric, we considered the duration during which the user was not in contact with the touch frame in SIT6 or was not holding the trigger in Scaled HOMER. Figure 5.2 presents this metric, recorded in seconds, with the results being as follows:

- **Task 1 (Ultra-Close Simple)**: No statistically significant difference in idle time between the two methods ($Z = -0.896, p = 0.370$).
- **Task 2 (Ultra-Close Complex)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($Z = -3.920, p < 0.001$), with $6.45 \pm 1.35s$ compared to $13.98 \pm 2.90s$.
- **Task 3 (Close Simple)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($t(19) = 5.017, p < 0.001$), with $6.93 \pm 1.06s$ compared to $8.97 \pm 1.51s$.
- **Task 4 (Close Complex)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($t(18) = 9.191, p < 0.001$), with $7.40 \pm 1.14s$ compared to $15.82 \pm 3.74s$.
- **Task 5 (Medium Simple)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($t(21) = 6.940, p < 0.001$), with $7.04 \pm 0.85s$ compared to $11.28 \pm 2.64s$.
- **Task 6 (Medium Complex)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($t(18) = 8.843, p < 0.001$), with $7.67 \pm 1.25s$ compared to $15.58 \pm 3.94s$.

- **Task 7 (Far Simple)**: No statistically significant difference in idle time between the two methods ($t(18) = -0.396, p = 0.697$).
- **Task 8 (Far Complex)**: Scaled HOMER had statistically significantly lower idle time than SIT6 ($Z = -3.883, p < 0.001$), with $6.87 \pm 1.06s$ compared to $12.97 \pm 4.51s$.

Interpretation

The results for the idle time metric closely follow those of task completion time, revealing that Scaled HOMER consistently exhibited statistically significantly less idle time compared to SIT6 in most tasks. Regarding the complex task results (tasks 2, 4, 6, and 8), the shorter idle time observed in Scaled HOMER can once again be attributed to the inherent intuitiveness of manipulation in this technique. Throughout the Scaled HOMER experiments, participants demonstrated a natural flow of movement and rarely released the trigger on the VR controller to reflect upon their next movement. Instead, they utilized this trigger feature primarily to reset their hand's position when their arm reached its full extension, which usually had minimal impact on idle time. On the other hand, the SIT6 experiments showed that despite most participants' familiarity with the technique's gesture dictionary, they frequently paused their movements to think about which axis they needed to rotate along, resulting in a substantial increase in task idle time, especially for individuals with less 3D perception and academic qualifications.

Regarding the simple task results, tasks 3 and 5, which require X-axis and Z-axis rotation, respectively, exhibit greater idle time in SIT6 compared to Scaled HOMER. We again believe this to be caused by rotations in incorrect axes, either by user mistake, occasional inaccuracies in the state detection system, or the touch frame's high sensitivity. These unintentional rotations increase task difficulty and cause tasks to require additional thinking process by the user, resulting in prolonged idle time.

However, we did not observe statistically significant differences between the two techniques regarding tasks 1 and 7. This result is probably because, during Y-axis rotation, participants using Scaled HOMER tended to release the trigger more frequently to reset their hand position and alleviate pressure on the wrist, which led to increased idle time in this technique for these tasks. Furthermore, as mentioned earlier, SIT6 prioritized the RotationY state in its state verification order, making it easier to trigger and reducing idle time while users attempted to perform Y-axis rotation.

5.2.1.3 Active Time

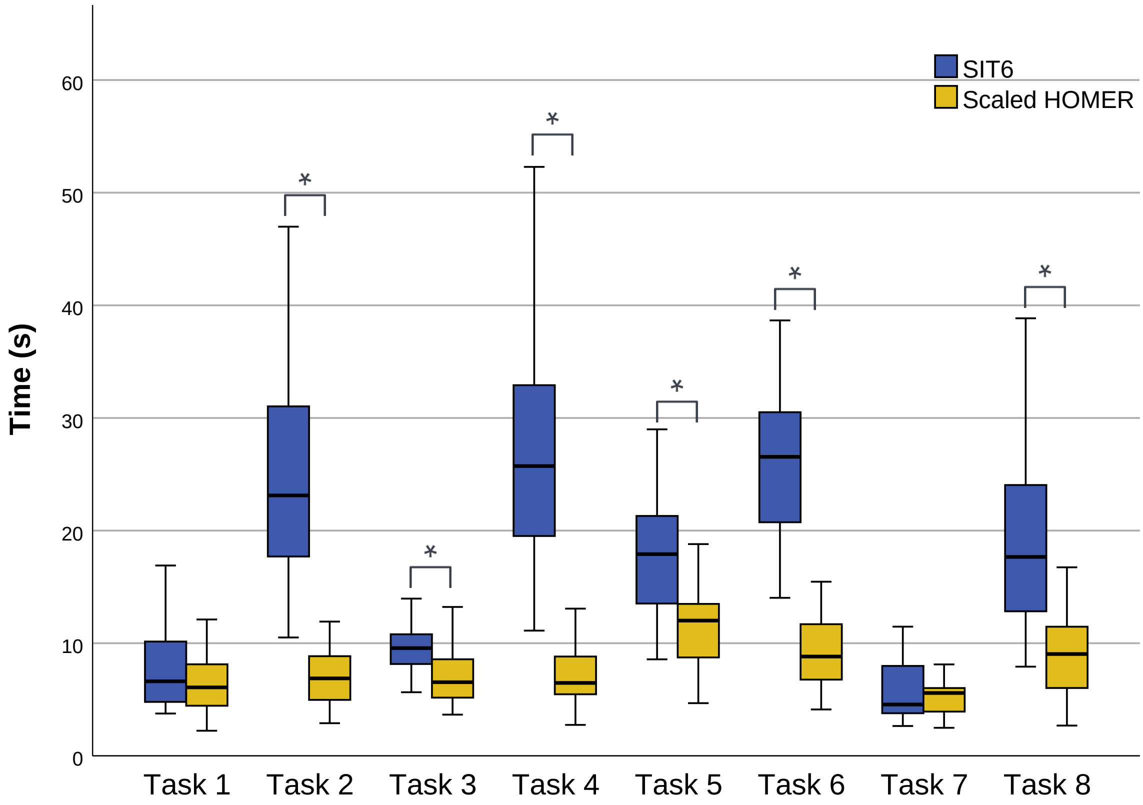


Figure 5.3: Box-plot of active time per task by technique.

In order to calculate the active time metric, we subtracted the idle time values from the completion time data for both techniques. Therefore, this metric measures the duration the user was in contact with the touch surface in SIT6 or held the VR controller's trigger in Scaled HOMER. The recorded metric, expressed in seconds, is presented in Figure 5.3 and provides the following results:

- **Task 1 (Ultra-Close Simple)**: No statistically significant difference in active time between the two methods ($Z = -0.629, p = 0.530$).
- **Task 2 (Ultra-Close Complex)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(23) = 8.666, p < 0.001$), with $6.83 \pm 2.54s$ compared to $24.15 \pm 8.76s$.
- **Task 3 (Close Simple)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(17) = 2.771, p = 0.013$), with $7.23 \pm 2.98s$ compared to $9.88 \pm 2.31s$.
- **Task 4 (Close Complex)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(21) = 8.602, p < 0.001$), with $7.57 \pm 2.96s$ compared to $26.85 \pm 10.28s$.
- **Task 5 (Medium Simple)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(21) = 3.644, p = 0.002$), with $11.56 \pm 3.72s$ compared to $17.52 \pm 5.87s$.

- **Task 6 (Medium Complex)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(23) = 11.413, p < 0.001$), with $9.30 \pm 3.33s$ compared to $26.18 \pm 7.15s$.
- **Task 7 (Far Simple)**: No statistically significant difference in active time between the two methods ($Z = -0.370, p = 0.711$).
- **Task 8 (Far Complex)**: Scaled HOMER had statistically significantly lower active time than SIT6 ($t(21) = 4.513, p < 0.001$), with $9.00 \pm 4.11s$ compared to $18.67 \pm 8.25s$.

Interpretation

Similar to the completion time and idle time metrics, the active time results indicate that Scaled HOMER achieved statistically significantly lower active times than SIT6 in most tasks. This result can again be attributed to Scaled HOMER's natural approach to manipulation, enabling users to intuitively and efficiently perform transformations. As a result, the overall total active time for the technique is reduced, particularly for the complex tasks 2, 4, 6, and 8, as observed in the study.

Regarding simple tasks 3 and 5, SIT6's higher active time values can be explained by the unintentional transformations caused by user errors or the touch frame's high sensitivity, as mentioned earlier. These accidental triggers of incorrect states significantly increase the difficulty of basic tasks, requiring additional touch movement and, consequently, more active time.

On the other hand, for simple tasks 1 and 7, we observe comparable active time values between SIT6 and Scaled HOMER. This outcome is possibly justified by the previously mentioned challenges associated with performing Y-axis rotation for Scaled HOMER. The physical strain imposed on the user during motion slows down the rotation movement, leading to increased active time for Scaled HOMER. Furthermore, since these two tasks only required a single Y-axis rotation and translation along the XZ-plane, which is the most straightforward transformation for SIT6 to trigger, the technique's active time was significantly reduced, matching the performance of Scaled HOMER.

5.2.1.4 Total Hand Movement

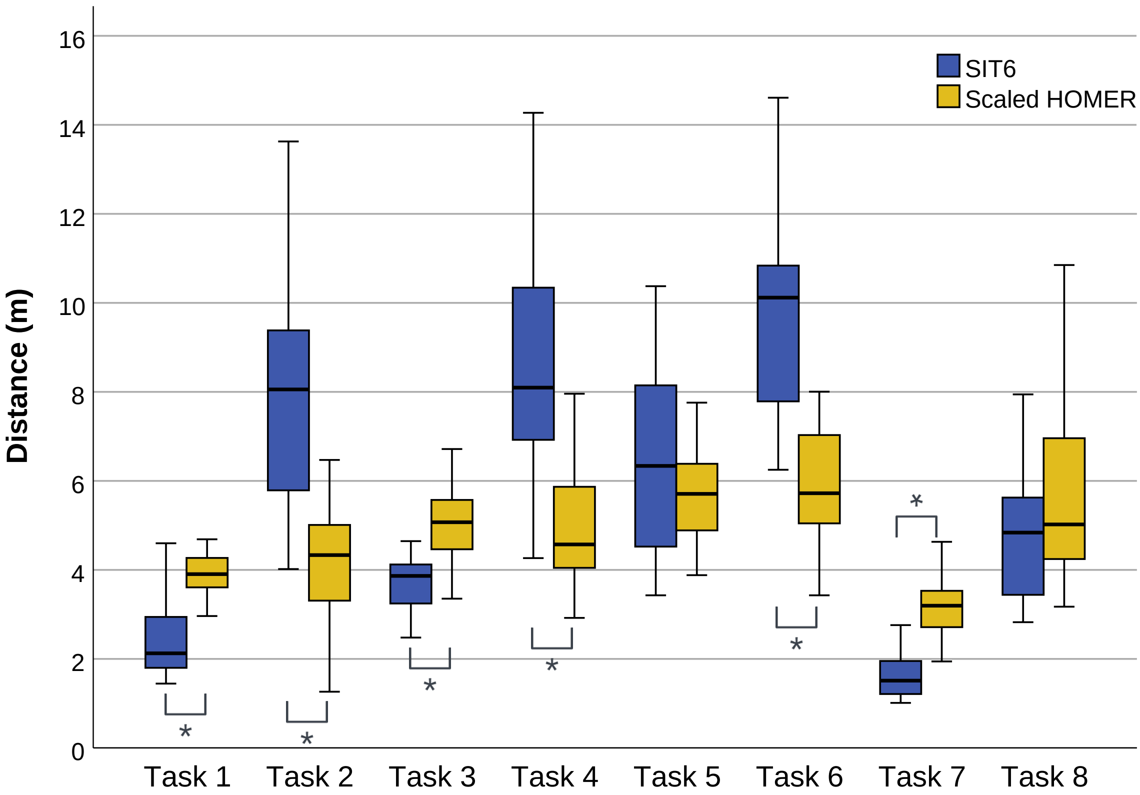


Figure 5.4: Box-plot of total hand movement per task by technique.

The total hand movement metric was also measured differently for the two techniques. For Scaled HOMER, we utilized the frame-by-frame controller positions to calculate hand movement. However, for SIT6, we had to estimate hand movement by converting the frame-by-frame touch positions, which were recorded in pixels, to real-life coordinates using the measurements of the touch frame. Figure 5.4 depicts the metric, recorded in meters, with the results being the following:

- **Task 1 (Ultra-Close Simple)**: SIT6 had statistically significantly lower hand movement than Scaled HOMER ($Z = -3.337, p = 0.001$), with $2.35 \pm 0.82m$ compared to $3.89 \pm 0.50m$.

- **Task 2 (Ultra-Close Complex)**: Scaled HOMER had statistically significantly lower hand movement than SIT6 ($t(22) = 5.555, p < 0.001$), with $4.27 \pm 1.30m$ compared to $7.85 \pm 2.39m$.

- **Task 3 (Close Simple)**: SIT6 had statistically significantly lower hand movement than Scaled HOMER ($t(19) = -4.473, p < 0.001$), with $3.70 \pm 0.60m$ compared to $5.06 \pm 1.00m$.

- **Task 4 (Close Complex)**: Scaled HOMER had statistically significantly lower hand movement than SIT6 ($t(18) = 5.808, p < 0.001$), with $4.90 \pm 1.24m$ compared to $8.66 \pm 2.33m$.

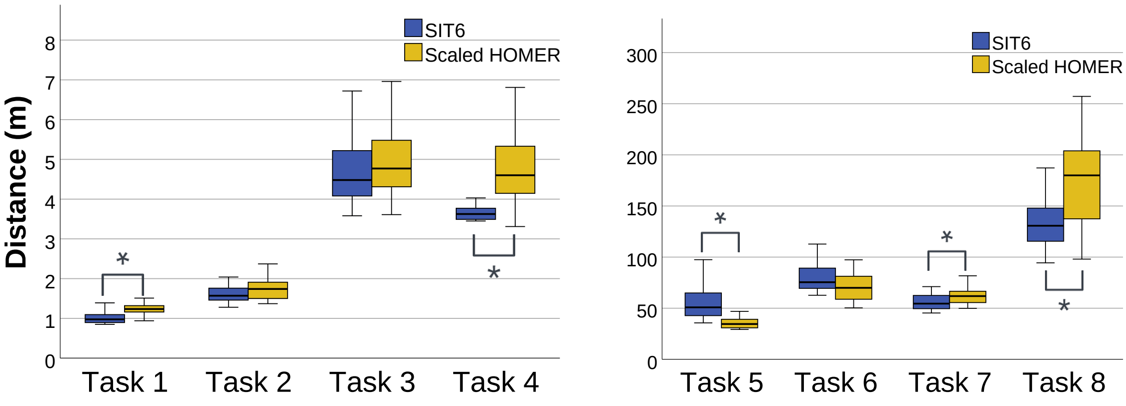
- **Task 5 (Medium Simple)**: No statistically significant difference in hand movement between the two methods ($t(22) = 1.793, p = 0.087$).
- **Task 6 (Medium Complex)**: Scaled HOMER had statistically significantly lower hand movement than SIT6 ($t(21) = 7.491, p < 0.001$), with $5.70 \pm 1.27m$ compared to $9.85 \pm 2.50m$.
- **Task 7 (Far Simple)**: SIT6 had statistically significantly lower hand movement than Scaled HOMER ($t(16) = -7.525, p < 0.001$), with $1.60 \pm 0.53m$ compared to $3.10 \pm 0.79m$.
- **Task 8 (Far Complex)**: No statistically significant difference in hand movement between the two methods ($t(17) = -1.147, p = 0.267$).

Interpretation

Upon examining the results for total hand movement, we found that SIT6 exhibited statistically significantly lower values of total hand movement in most simple tasks (1, 3, and 7) compared to Scaled HOMER. The observed differences in total hand movement during the simple tasks can be attributed to the efficient input mapping employed in our solution. With SIT6, users could easily translate objects by simply swiping a finger, resulting in shorter hand movement distances compared to the mid-air motions required in Scaled HOMER. The only exception to this trend is task 5, where both techniques demonstrated comparable results in terms of total hand movement. This particular result could possibly be associated with the nature of the task itself, as the difficulty in perceiving the depth of the docking point during this task frequently led users in SIT6 to drag the object farther than necessary, resulting in additional hand movement compared to the other simple tasks.

On the other hand, for complex tasks 2, 4, and 6, we concluded that Scaled HOMER had statistically significantly less hand movement than SIT6. From what we observed during the experiments, this significant increase in hand movement for SIT6 compared to the simple tasks is caused by the motion from multiple rotation gestures. In complex tasks, which involve rotation along multiple axes, users often resorted to trial and error to determine the correct axis for rotation. This approach significantly increased hand movement for SIT6 as users made multiple attempts to find the right axis. Conversely, Scaled HOMER did not encounter this issue as the object seamlessly rotated with the user's hand, minimizing the need for additional movement and resulting in more efficient interaction. Task 8 was the only complex task where both techniques demonstrated comparable results. This outcome could be linked to two factors: firstly, the task could have allowed users to visualize the required axes for rotation more easily than other complex tasks, leading to a more efficient performance in SIT6. Secondly, since task 8 was the final task of the sequence, users could have already gained experience and familiarity with SIT6, improving their performance and leading to less total hand movement.

5.2.1.5 Total Object Translation



(a) (b)

Figure 5.5: Box-plot of total object translation per task by technique, split into two graphs due to scale differences. (a) Tasks 1-4. (b) Tasks 5-8.

The total object translation metric for SIT6 was obtained by summing the total distances the object translated in the *TranslationXZ* and *TranslationY* states. In the case of Scaled HOMER, to exclude accidental translation movements during controller rotation, only object position differences per frame above a threshold value were considered translations. The selected value for our experiments was 0.001, which was chosen diligently based on the VR controller's sensitivity. The data for this metric, which is measured in meters, is presented in Figure 5.5 and yields the following results:

- **Task 1 (Ultra-Close Simple)**: SIT6 had statistically significantly lower object translation than Scaled HOMER ($Z = -2.749, p = 0.006$), with $1.02 \pm 0.16m$ compared to $1.23 \pm 0.14m$.
- **Task 2 (Ultra-Close Complex)**: No statistically significant difference in object translation between the two methods ($t(17) = -1.126, p = 0.276$).
- **Task 3 (Close Simple)**: No statistically significant difference in object translation between the two methods ($t(21) = -0.486, p = 0.632$).
- **Task 4 (Close Complex)**: SIT6 had statistically significantly lower object translation than Scaled HOMER ($Z = -3.623, p < 0.001$), with $3.66 \pm 0.18m$ compared to $4.83 \pm 0.96m$.
- **Task 5 (Medium Simple)**: Scaled HOMER had statistically significantly lower object translation than SIT6 ($Z = -3.574, p < 0.001$), with $35.65 \pm 5.57m$ compared to $56.08 \pm 16.88m$.
- **Task 6 (Medium Complex)**: No statistically significant difference in object translation between the two methods ($Z = -1.686, p = 0.092$).

- **Task 7 (Far Simple)**: SIT6 had statistically significantly lower object translation than Scaled HOMER ($t(15) = -2.316, p = 0.035$), with $56.03 \pm 7.62m$ compared to $63.98 \pm 10.59m$.
- **Task 8 (Far Complex)**: SIT6 had statistically significantly lower object translation than Scaled HOMER ($t(18) = -3.493, p = 0.003$), with $126.74 \pm 22.69m$ compared to $170.05 \pm 45.44m$.

Interpretation

Unlike the previous metrics, the total object translation results did not exhibit explicit trends across the simple and complex task groups. These results aligned with our expectations, as task complexity was evaluated based on the number of axes of rotation required rather than the difficulty of the necessary translations.

In most cases, SIT6 achieved task completion with either comparable or lower total object translation movement than Scaled HOMER. Specifically, SIT6 demonstrated statistically significantly lower total object translation values than Scaled HOMER in tasks 1, 4, 7, and 8. This outcome can be attributed to the precise manipulation enabled by our solution's input mapping, allowing users to perform accurate movements at short and long distances. In contrast, experiments revealed that Scaled HOMER's input scaling, combined with the non-separation of degrees of freedom, often led to unintended translations while users attempted to rotate objects. This issue was further exacerbated during out-of-reach manipulation (e.g., tasks 7 and 8) due to the technique's distance-based scaling, increasing total object translation movement.

In tasks 2, 3, and 6, where the objects were manipulated at a closer distance to the user, Scaled HOMER's out-of-reach scaling problem had a lesser impact on increasing the object's translation movement. As a result, the total object translation movement in Scaled HOMER remained comparable to that of SIT6.

Finally, the only task where Scaled HOMER showed statistically significantly less object translation movement than SIT6 was task 5, an outcome that can be attributed to the specific design of the task itself, as discussed previously. In SIT6, participants tended to unintentionally drag the object farther than necessary in this task due to depth perception difficulties, leading to unnecessary object translation and resulting in a higher total object translation movement compared to Scaled HOMER.

5.2.2 Subjective Results

For the analysis of the qualitative results, we assumed a non-normal distribution of the data. Consequently, we used the Wilcoxon Signed Ranks test to assess whether there were any statistically significant differences between the methods, which provided the following results:

- **Question 1**: SIT6 had statistically significantly higher value answers than Scaled HOMER ($Z = -2.277, p = 0.023$), with 4.42 ± 0.76 compared to 3.81 ± 1.06.
- **Question 2**: Scaled HOMER had statistically significantly higher value answers than SIT6 ($Z = -4.095, p < 0.001$), with 3.62 ± 1.27 compared to 1.50 ± 0.65.
- **Question 3**: No statistically significant difference in answer values between the two methods ($Z = -1.852, p = 0.064$).
- **Question 4**: No statistically significant difference in answer values between the two methods ($Z = -0.294, p = 0.769$).
- **Question 5**: Scaled HOMER had statistically significantly higher value answers than SIT6 ($Z = -2.996, p = 0.003$), with 4.69 ± 0.55 compared to 4.08 ± 0.80.
- **Question 6**: SIT6 had statistically significantly higher value answers than Scaled HOMER ($Z = -3.018, p = 0.003$), with 4.81 ± 0.49 compared to 3.85 ± 1.22.

Further results of this analysis for each technique, such as the median and interquartile range (IQR) are presented in Table 5.1.

Question	Median (IQR)	
	SIT6	**Scaled HOMER**
Q1 - Comfortable *	5 (1)	4 (2)
Q2 - Physically demanding *	1 (1)	4 (1.25)
Q3 - Mentally demanding	3 (1)	2 (2)
Q4 - Easy to use	4 (2)	4 (1.25)
Q5 - Easy to learn *	4 (2)	5 (1)
Q6 - Suitable for DeskVR *	5 (0)	4 (2)

Table 5.1: Median and interquartile range values for the answers of each question by technique. * indicates statistically significant differences between the methods.

Interpretation

Regarding the first two questions, participants' answers showed that SIT6 was statistically significantly more comfortable and less physically demanding than Scaled HOMER. These outcomes were anticipated, given that SIT6 was designed as a touch-based object manipulation technique, an interaction paradigm chosen to minimize exhaustion and improve comfort in our solution. On the other hand, the user experiments demonstrated that Scaled HOMER consistently placed considerable strain on the user's arm and wrist during transformations, resulting in a noticeably less comfortable experience.

The responses to questions 3 and 4 revealed that both techniques were comparable in terms of mental demand and ease of use. While most users could memorize SIT6's gesture dictionary without much difficulty, they occasionally encountered problems while selecting the correct axis for rotation during tasks. This rotation issue was less frequent for Scaled HOMER, as users generally encountered minimal difficulty determining the direction in which they needed to move their hand. Consequently, we initially anticipated that the separation of degrees of freedom and the resulting perception problems in 3D space would negatively impact the results of both questions regarding our solution. However, contrary to our expectations, these factors did not substantially influence the results. The observed outcome could potentially be attributed to the negative influence of physical demand on the user's perception of Scaled HOMER's ease of use. Additionally, some users may have found the technique mentally demanding due to the inability to isolate transformations effectively, with users frequently unintentionally performing rotations while attempting translations and vice versa.

In question 5, responses demonstrated that Scaled HOMER was statistically significantly easier to learn than SIT6. This result was expected since Scaled HOMER possesses an intuitive motion intrinsic to natural hand movement, causing users to adapt to the technique considerably faster than SIT6, as seen in the user experiments. Conversely, SIT6 required users to effectively learn and memorize a gesture dictionary, which presented difficulties and slowed down the adaptation process for the technique.

For the last question, the users' answers exhibited SIT6 as a statistically significantly more suitable technique for DeskVR than Scaled HOMER. Once again, this development was anticipated as our solution was intentionally designed for a seated-desk scenario, adapting to space constraints. In contrast, during the Scaled HOMER experiments, users frequently encountered problems such as accidentally striking the desk with the VR controller, indicating a lack of physical space to execute the required motions for the technique. Nevertheless, many participants suggested that the technique would be suitable for performing the tasks if sufficient space were available.

5.2.3 Discussion

After analyzing and interpreting the outcomes for each objective and subjective metric, we can now take a comprehensive view of the overall results with the goal of finding further insights and clarify the connections between the various metrics.

First, upon analyzing the results for completion time, idle time, and active time, it becomes apparent that these three metrics exhibit a consistent pattern, with SIT6 consistently demonstrating statistically significantly higher values than Scaled HOMER across most tasks (2, 3, 4, 5, 6 and 8). As mentioned earlier, in our approach, participants spent a significant amount of time engaged in the thought process of axis selection during rotations or making repeated attempts when gestures failed to activate the intended state, therefore raising the total idle time. However, this increase in idle time did not result in lower active time for SIT6 relative to Scaled HOMER. This outcome is possibly caused by users inadvertently activating the incorrect rotation mode, requiring them to perform more movements than initially anticipated for the task and leading to a substantial rise in active time. In the case of Scaled HOMER, the lower idle time values can be attributed to the simplicity of initiating and executing movement, where users only needed to press the trigger and move the VR controller. Moreover, the natural object manipulation offered by Scaled HOMER enabled users to complete tasks more efficiently, reducing overall active time. This contrast in difficulty level between the techniques when initiating and performing movement, especially rotations, which probably led to the statistically significant differences observed in idle and active times between the techniques, consequently caused the statistically significant higher completion time values for SIT6. Aditionally, these notable difficulties while using the SIT6 technique, further amplified by limitations in the state detection system and the touch frame's high sensitivity, hindered the participant's initial adaptation to the technique, as observed in the experiments. Thus, these time results corroborate the subjective metric result that indicated Scaled HOMER was easier to learn than SIT6.

Next, we can see that, for most complex tasks, SIT6 has statistically significantly more total hand movement than Scaled HOMER. However, participants' feedback indicated that SIT6 was the more comfortable and less physically demanding technique. These contradicting results from the objective and subjective metrics are likely explained by the fact that while SIT6 may involve more hand movement in specific tasks, its gestures are executed in just two dimensions and allow for resting the hands on the desk. On the other hand, Scaled HOMER requires movements to be performed in three dimensions, demanding the user to elevate the controller in the air, which places additional strain on the user and contributes to the perceived discomfort, despite the potential for reduced hand movement. Thus, it is inappropriate to directly compare the required physical effort for a given total of hand movement between the two techniques.

5.2.3.1 Research Questions and Hypothesis

With all the results interpreted and discussed, it is now possible to answer the proposed research questions:

RQ1: Can SIT6 have an equal or greater success rate at object manipulation tasks compared to a state-of-the-art mid-air baseline with manipulation capabilities for any distance? Yes. The objective results showed a 100% task success rate for both SIT6 and Scaled HOMER, indicating that both techniques were equally effective in enabling users to accomplish the assigned tasks. Furthermore, as the tasks were subject to a 2-minute time limit and the majority of participants were able to finish each task within that timeframe comfortably, it is evident that both techniques can complete object manipulation tasks within a reasonable time frame.

RQ2: Can SIT6 be faster at completing object manipulation tasks than a state-of-the-art mid-air baseline with manipulation capabilities for any distance? The results demonstrate that, in general, no. Scaled HOMER was statistically significantly faster than SIT6 for most tasks, especially those requiring more complex rotations involving multiple axes. Nevertheless, SIT6 demonstrates comparable performance to Scaled HOMER in specific simple tasks, mainly those where rotation can be easily executed.

RQ3: Can SIT6 complete object manipulation tasks while being less physically demanding than a state-of-the-art mid-air baseline with manipulation capabilities for any distance? Yes. From the qualitative results, we can conclude that SIT6 was statistically significantly more comfortable and less physically demanding than Scaled HOMER. This outcome also holds in tasks where our solution demands more hand movement than Scaled HOMER. Since this additional movement comes from the additional rotation motion and is performed in two dimensions, it is still less physically exhausting than the three-dimensional movement required in Scaled HOMER.

With the research questions answered, we can now validate or refute the hypothesis:

Touch-based object manipulation techniques can be as effective as mid-air object manipulation in a DeskVR scenario while being less physically demanding.

Overall, regarding efficiency, SIT6 did perform slower than Scaled HOMER. The main factor contributing to this performance difference is likely the difficulty involved in executing rotations in our technique. Although SIT6 incorporates DOF separation, which can enhance rotation performance in touch-based techniques, as seen in the approach developed by Veit et al. [36], it also introduces additional complexities when the manipulable object is viewed from challenging perspectives. In contrast to the experiments conducted by Veit et al., which involved rotating a stationary object in front of the user, our docking tasks often required moving the object to final positions that were not centered with the user's camera. As a result, difficulties frequently arose when selecting the rotation axis in those positions, requiring participants to possess adequate 3D perception skills, which the average user may not have. Notably, another factor that impacted the performance of SIT6 was the need for precise movements to trigger certain transformations, particularly rotations, which occasionally proved challenging to execute.

Despite that, in terms of effectiveness, SIT6 did manage to complete every single task across all participants while being less physically demanding than Scaled HOMER. These results prove that although our solution was not as efficient at most object manipulation tasks, it was consistently as effective as the mid-air baseline. Therefore, the proposed hypothesis is **validated**.

An important final remark to consider in our approach's overall results is that while the technique exhibited lower task efficiency and posed a steeper learning curve for the participants we tested (average users), SIT6 was specifically designed for professional settings where extended learning and training programs are common to master the utilized software. Considering this, we believe that the technique would demonstrate notably better efficiency in a professional context with qualified individuals, which, along with the comfort and minimal physical demands already demonstrated in the results, solidifies the notion that our proposed approach is suitable for prolonged and productive DeskVR interactions.

5.3 Summary

In this chapter, we analyzed and interpreted the data obtained from user experiments. Section 5.1 defined the subjective and objective metrics used in our analysis and the filtering process employed to obtain them.

Section 5.2 explained the data analysis process and statistical tests conducted on the selected objective and subjective metrics. For the objective metrics, outliers were first removed, followed by applying the Shapiro-Wilk normality test. A paired-samples t-test was used for normally distributed data, while the Wilcoxon signed-ranked test was employed for non-normal data. A non-normal distribution was assumed for the subjective metrics, and thus only the Wilcoxon signed-ranked test was used. The section then provided a detailed breakdown and interpretation of the results for each metric. It began with the objective metrics, which included completion time, idle time, active time, total hand movement, and total object translation. Subsequently, the section addressed the subjective metrics, analyzing the results question by question. Following the analysis of individual metrics, the section discussed the overall outcome, drawing connections between the results and the observed factors during the user experiments, which provided insights into the implications and significance of the findings. Finally, with all the necessary information retrieved, the section addressed the three research questions posed earlier, allowing for the validation of the hypothesis. The results ultimately supported the notion that touch-based object manipulation techniques, although less efficient in task completion, can be equally effective as mid-air object manipulation in a DeskVR scenario while also being less physically demanding.

Chapter 6

Conclusions

Virtual Reality (VR) provides users with immersive experiences and improved visualization capabilities, potentially benefiting various professional settings such as architecture, content creation, data analysis, and medicine. However, the utilization of VR for extended periods is limited by the need for physically demanding mid-air movements, a problem that DeskVR addresses by allowing users to fully immerse themselves in VR while sitting at a desk, enabling seamless integration into their workflow. Nevertheless, designing suitable techniques for this context is challenging due to limited physical mobility and space. With this in mind, this dissertation focused on proposing a novel object manipulation method tailored for DeskVR, employing undemanding and intuitive gestures while also addressing precision and out-of-reach manipulation issues.

From the literature study of existing object manipulation techniques, we determined that our proposed solution had to have 6-DOF and incorporate an indirect, touch-based paradigm while using distance and velocity-based input mappings adapted from mid-air approaches. This decision led to the hypothesis that touch-based object manipulation techniques could be as effective as mid-air object manipulation in a DeskVR scenario while being less physically demanding. Consequently, we developed Scaled Indirect Touch 6-DOF, or SIT6, an object manipulation technique with a gesture dictionary inspired by Indirect6 and which incorporates scaling approaches from Scaled HOMER. To implement this solution, we developed a state machine that accurately interprets performed gestures and translates them into specific states, containing several error-handling mechanisms. Considering our technique had DOF separation, we also included visual indicators to assist users in object manipulation tasks.

Subsequently, we proceeded with user experiments to assess the effectiveness of our solution compared to the chosen baseline: Scaled HOMER. The experimental process involved performing various docking tasks, which consisted of moving an object from one location to another. These tasks were designed with varying difficulty levels and were categorized based on the distance between the user and the object/docking point, which enabled us to evaluate the performance of the techniques across different distance scenarios. To conclude the procedure, participants were given a questionnaire for each tested technique to provide feedback on factors such as comfort, physical and mental demand, ease of use, ease of learning, and suitability for DeskVR.

As for the results, we found that SIT6 showed longer task completion times than Scaled HOMER in most tasks. However, when considering task success rate and user feedback, we discovered that SIT6 was equally effective as the mid-air approach while being less physically demanding. Therefore, our hypothesis has been validated, demonstrating that while our technique may be less efficient, it remains suitable for DeskVR scenarios without compromising effectiveness.

6.1 Future Work

Although the research questions have been addressed and the proposed hypothesis has been validated, certain aspects of our solution still require additional investigation, exploration, and testing. This study explored the potential of a touch-based object manipulation technique designed explicitly for DeskVR environments while addressing precision and out-of-reach manipulation issues. The proposed solution was evaluated against a baseline mid-air approach, testing both techniques' effectiveness and efficiency.

In order to enhance the evaluation of the user experiments, one aspect that could be improved is the inclusion of a greater number of tasks with a wider diversity of distances and rotations within the experimental sequence. While this would extend the duration of the experiments, it would allow the techniques to be tested in a broader range of scenarios, offering more clarity regarding the contexts in which each technique performs best. Additionally, while the technique's gesture dictionary has been thoroughly analyzed in the results, there is still room for a more in-depth study of the influence of the employed input mapping. This examination could be achieved by testing different scenarios with varying scaling coefficients, providing a better understanding of the effects of such mapping values. Likewise, conducting additional tests to assess the impact of the implemented visual indicators also holds significant potential for valuable insights.

Moreover, although our solution has comparable effectiveness to the selected baseline, there is room for further improvement in terms of efficiency. One way to improve efficiency could be by rethinking the approach's gesture dictionary for the rotation movements, as this was the area where users encountered the most challenges. By incorporating rotation gestures that are more distinct and easier to execute, users would spend less time trying to repeatedly engage in rotation movement. Furthermore, enhancements can be made to some parts of our solution's implementation, such as the state machine and the gesture detection controller. By optimizing these components, our system would be more effective at detecting motion, reducing the need for repeated attempts to trigger a state when a user's gesture is not executed perfectly. Consequently, this would enhance the overall speed and fluidity of the technique, increasing task completion efficiency. Similarly, upgrading the touch-detection equipment to a touch surface with lower sensitivity and higher precision than the currently used infrared multi-touch frame would significantly reduce accidental user inputs, another improvement that would greatly increase efficiency.

Lastly, while our solution provides users with object manipulation in 6 DOFs, enabling translation and rotation along every axis, there is an opportunity for design changes to expand its functionality to include object-scaling capabilities. Implementing this feature would likely require introducing additional gestures, increasing the complexity of the technique's motion dictionary. However, it would further enhance the versatility and flexibility of our solution, enabling users to interact with virtual objects in a more customizable manner.

References

[1] Oscar Kin-Chung Au, Chiew-Lan Tai, and Hongbo Fu. Multitouch Gestures for Constrained Transformation of 3D Objects. *Computer Graphics Forum*, 31(2pt3):651–660, May 2012.

[2] Chris Auteri, Mark Guerra, and Scott Frees. Increasing Precision for Extended Reach 3D Manipulation. *International Journal of Virtual Reality*, 12(1):66–73, January 2013. Number: 1.

[3] Hrvoje Benko and Steven Feiner. Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection. In *2007 IEEE Symposium on 3D User Interfaces*, March 2007.

[4] Björn Bollensdorff, Uwe Hahne, and Marc Alexa. The effect of perspective projection in multi-touch 3D interaction. In *Proceedings of Graphics Interface 2012*, GI '12, pages 165–172, CAN, May 2012. Canadian Information Processing Society.

[5] Benoît Bossavit, Asier Marzo, Oscar Ardaiz, Luis Diaz De Cerio, and Alfredo Pina. Design Choices and Their Implications for 3D Mid-Air Manipulation Techniques. *Presence*, 23(4):377–392, November 2014. Conference Name: Presence.

[6] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 35–ff., New York, NY, USA, April 1997. Association for Computing Machinery.

[7] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., USA, 2004.

[8] Isaac Cho and Zachary Wartell. Evaluation of a bimanual simultaneous 7DOF interaction technique in virtual environments. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 133–136, March 2015.

[9] Aurélie Cohé, Fabrice Decle, and Martin Hachet. tBox: A 3D Transformation Widget designed for Touch-screens. *ACM CHI Conference on Human Factors in Computing Systems, [Note]*, May 2011.

[10] Bruno R. De Araújo, Géry Casiez, Joaquim A. Jorge, and Martin Hachet. Mockup Builder: 3D modeling on and above the surface. *Computers & Graphics*, 37(3):165–178, May 2013.

[11] Scott Frees, G. Drew Kessler, and Edwin Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction*, 14(1):2–es, May 2007.

[12] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3d interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 1147–1156, New York, NY, USA, 2007. Association for Computing Machinery.

[13] Mark Hancock, Thomas ten Cate, and Sheelagh Carpendale. Sticky tools: Full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, page 133–140, New York, NY, USA, 2009. Association for Computing Machinery.

[14] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pages 139–148, New York, NY, USA, October 2009. Association for Computing Machinery.

[15] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124, CAN, May 2009. Canadian Information Processing Society.

[16] Kiyoshi Kiyokawa, Haruo Takemura, and Naokazu Yokoya. Manipulation aid for two-handed 3-d designing within a shared virtual environment. In Michael J. Smith, Gavriel Salvendy, and Richard J. Koubek, editors, *Design of Computing Systems: Social and Ergonomic Considerations, Proceedings of the Seventh International Conference on Human-Computer Interaction, (HCI International '97), San Francisco, California, USA, August 24-29, 1997, Volume 2*, pages 937–940. Elsevier, 1997.

[17] Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. *Computer Graphics Forum*, 31(7pt1):2047–2055, September 2012.

[18] Daniel P. Mapes and J. Michael Moshell. A two-handed interface for object manipulation in virtual environments. *Presence: Teleoperators and Virtual Environments*, 4(4):403–416, January 1995.

[19] Anthony Martinet, Géry Casiez, and Laurent Grisoni. The effect of DOF separation in 3D manipulation tasks with multi-touch displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pages 111–118, New York, NY, USA, November 2010. Association for Computing Machinery.

[20] Daniel Mendes, Ariel Caputo, Andrea Giachetti, Alfredo Ferreira, and Joaquim Jorge. A Survey on 3D Virtual Object Manipulation: From the Desktop to Immersive Virtual Environments: Survey on 3D Virtual Object Manipulation. *Computer Graphics Forum*, 38, April 2018.

[21] Daniel Mendes, Pedro Lopes, and Alfredo Ferreira. Hands-on interactive tabletop lego application. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE '11, New York, NY, USA, 2011. Association for Computing Machinery.

[22] João Monteiro. *Object Selection in DeskVR*. M.Sc dissertation, Faculdade de Engenharia da Universidade do Porto, Porto, July 2022.

[23] Thi-Thuong Huyen Nguyen, Thierry Duval, and Charles Pontonnier. *A New Direct Manipulation Technique for Immersive 3D Virtual Environments*. The Eurographics Association, 2014. Accepted: 2014-12-17T13:12:34Z ISSN: 1727-530X.

[24] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, I3D '99, pages 141–145, New York, NY, USA, April 1999. Association for Computing Machinery.

[25] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 79–80, New York, NY, USA, November 1996. Association for Computing Machinery.

[26] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. A screen-space formulation for 2D and 3D direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pages 69–78, New York, NY, USA, October 2009. Association for Computing Machinery.

[27] Denise Rey and Markus Neuhäuser. Wilcoxon-signed-rank test. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*, pages 1658–1659. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[28] Amanda Ross and Victor L. Willson. Paired samples t-test. In *Basic and Advanced Statistical Tests: Writing Results Sections and Creating Tables and Figures*, pages 17–19. SensePublishers, Rotterdam, 2017.

[29] Samuel S. Shapiro and Martin B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, December 1965.

[30] Ken Shoemake. ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the conference on Graphics interface '92*, pages 151–156, San Francisco, CA, USA, September 1992. Morgan Kaufmann Publishers Inc.

[31] Adalberto L. Simeone. Indirect touch manipulation for interaction with stereoscopic displays. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 13–22, 2016.

[32] Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1297–1306, New York, NY, USA, May 2012. Association for Computing Machinery.

[33] Mauricio Sousa, Daniel Mendes, Soraia Paulo, Nuno Matela, Joaquim Jorge, and Daniel Simões Lopes. Vrrrroom: Virtual reality for radiologists in the reading room. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 4057–4062, New York, NY, USA, 2017. Association for Computing Machinery.

[34] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, USA, May 1995. ACM Press/Addison-Wesley Publishing Co.

[35] Sven Strothoff, Dimitar Valkov, and Klaus Hinrichs. Triangle cursor: interactions with objects above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 111–119, New York, NY, USA, November 2011. Association for Computing Machinery.

[36] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. Influence of degrees of freedom's manipulation on performances during orientation tasks in virtual reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pages 51–58, New York, NY, USA, November 2009. Association for Computing Machinery.

[37] Yanqing Wang, Christine L. MacKenzie, Valerie A. Summers, and Kellogg S. Booth. The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, page 312–319, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[38] Curtis Wilkes and Doug A. Bowman. Advantages of velocity-based scaling for distant 3D manipulation. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, VRST '08, pages 23–29, New York, NY, USA, October 2008. Association for Computing Machinery.

[39] Siju Wu, Amine Chellali, Samir Otmane, and Guillaume Moreau. Touchsketch: A touch-based interface for 3d object manipulation and editing. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, VRST '15, page 59–68, New York, NY, USA, 2015. Association for Computing Machinery.

[40] Daniel Zielasko, Benjamin Weyers, Martin Bellgardt, Sebastian Pick, Alexander Meibner, Tom Vierjahn, and Torsten W. Kuhlen. Remain seated: towards fully-immersive desktop VR. In *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*, pages 1–6, March 2017.

Appendix A

Technique Evaluation Questionnaire

The technique evaluation form was designed to gather user feedback on each approach and comprised six statements. Users were asked to rate their level of agreement with each statement using a 5-point Likert Scale, ranging from 1 ("Strongly Disagree") to 5 ("Strongly Agree"). The questionnaire included the following statements:

- The technique was comfortable to use.
- The technique was physically demanding.
- The technique was mentally demanding.
- The technique was easy to use.
- The technique was easy to learn.
- The technique was suitable for the task at hand.

Appendix B

Controller Flowchart

The flowchart diagram depicted in Figure B.1 illustrates the complete detection process that the controller executes to link a specific state to the performed gesture, offering a comprehensive overview of the step-by-step procedure.

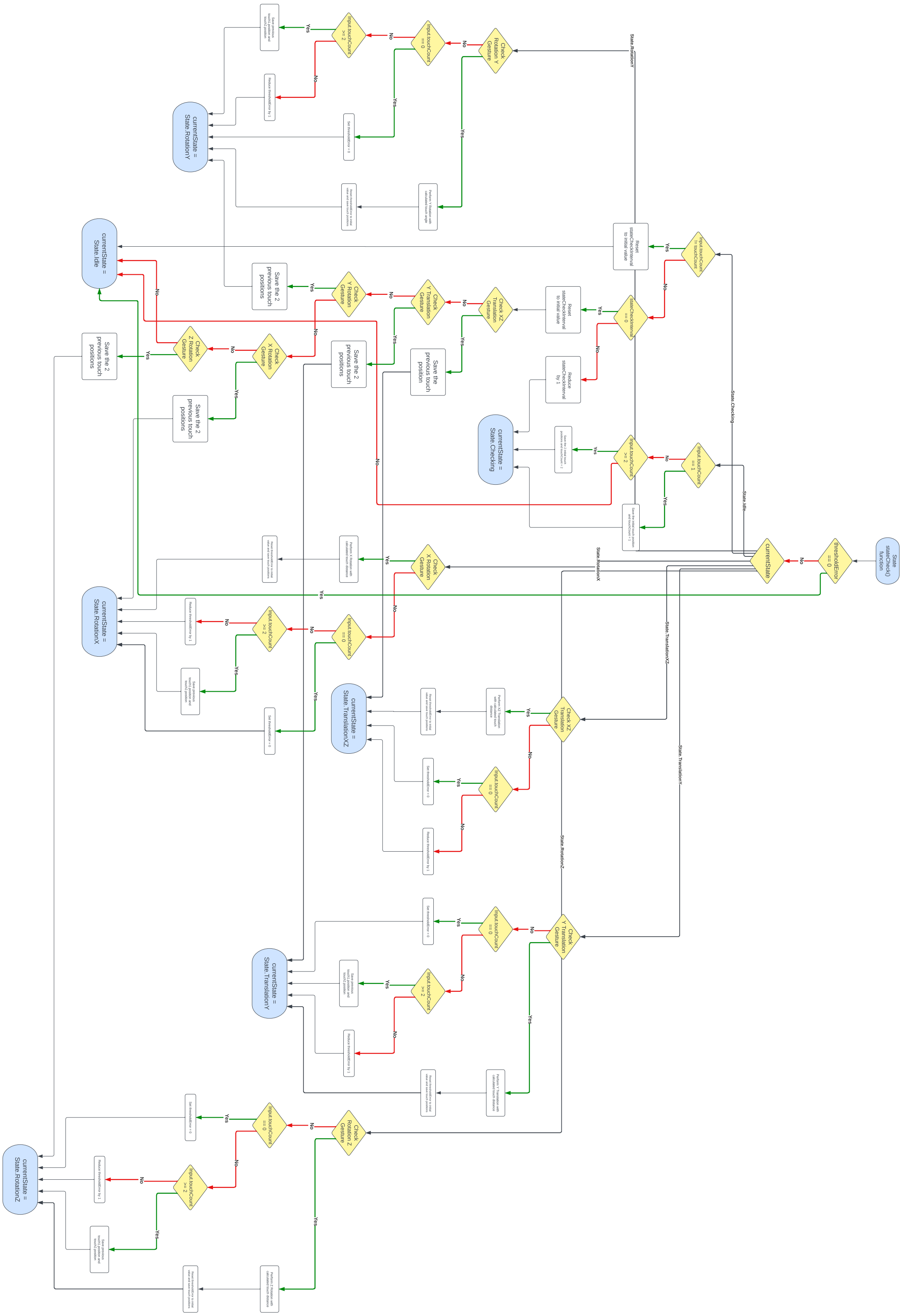


Figure B.1: Flowchart of SIT6's controller.