Universidade de Lisboa

Instituto Superior Técnico

Manipulation of 3D Objects in Immersive Virtual Environments

Daniel Filipe Martins Tavares Mendes

Supervisor: Doctor Alfredo Manuel dos Santos Ferreira Júnior

Co-Supervisor: Doctor Joaquim Armando Pires Jorge

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Jury final classification: Pass with Distinction and Honour

2018

Universidade de Lisboa

Instituto Superior Técnico

Manipulation of 3D Objects in Immersive Virtual Environments

Daniel Filipe Martins Tavares Mendes

Supervisor: Doctor Alfredo Manuel dos Santos Ferreira Júnior

Co-Supervisor: Doctor Joaquim Armando Pires Jorge

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Jury final classification: Pass with Distinction and Honour

Jury

Chairperson: Doctor Luís Eduardo Teixeira Rodrigues, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor Maria Beatriz Alves de Sousa Santos, Universidade de Aveiro

Doctor Rui Filipe Fernandes Prada, Instituto Superior Técnico, Universidade de Lisboa

Doctor Maximino Esteves Correia Bessa, Escola de Ciências e Tecnologia, Universidade de Trás-os-Montes e Alto Douro

Doctor Carlos Alberto Pacheco dos Anjos Duarte, Faculdade de Ciências, Universidade de Lisboa

Doctor Alfredo Manuel dos Santos Ferreira Júnior, Instituto Superior Técnico, Universidade de Lisboa

Funding Institutions

Fundação para a Ciência e a Tecnologia (SFRH/BD/91372/2012)

2018

Abstract

Object manipulation is a key feature in most virtual environments (VE), including immersive VE. The spatial input typically associated with these immersive VE can offer natural metaphors, allowing users to directly grab, move and rotate objects in a similar way to how it is done in the physical world, making manipulations feel more natural. However, mid-air gestures compromise manipulation accuracy, whether due to limitations in tracking solutions or the human dexterity itself.

In this research, we aimed at reducing the impact of the lack of precision when interacting with 3D objects in mid-air within immersive VE. After surveying the state-of-the art on object manipulation following several interaction paradigms, we began by assessing the performance of existing mid-air manipulation techniques in both semi and fully-immersive VE through user evaluations. Our findings suggest that, if no restrictions exist, the best approach is to use exact mappings, within a fully-immersive environment. Focusing on increasing the precision of object manipulation in mid-air, we investigated if it can benefit from separating degrees-of-freedom (DOF), which has been proved useful in other interaction paradigms. Our results showed that single DOF control can improve precision at the cost of additional time for complex tasks. To test if it can be combined with scaled translations for increased precision, and if custom transformation axes could perform faster than using axes from fixed frames, we developed two novel manipulation techniques. We found that no significant improvements came from scaling down isolated translations, and that users favored 3-DOF manipulations above all, while keeping translation and rotation independent. The accuracy of mid-air gestures also play an important role in selecting objects outside arms' reach. We developed a new selection technique for immersive VE, which combines cone-casting with an iterative progressive refinement strategy, teleporting users closer to intersected objects. Results of a user evaluation revealed our technique as a versatile approach to out-of-reach target acquisition, combining accurate selection with consistent times across different scenarios.

In conclusion, we have validated our thesis, which states that DOF separation and iterative progressive refinement strategies can be successfully used to provide more effective mid-air interactions within immersive virtual environments.

Resumo

A manipulação de objetos é uma tarefa fundamental na maioria dos ambientes virtuais (AV), incluindo AV imersivos. A entrada espacial tipicamente associada a AV imersivos oferece metáforas naturais, permitindo que os utilizadores agarrem, movam e rodem objetos semelhantemente ao que fazem no mundo físico, tornando as manipulações mais naturais. No entanto, gestos no ar comprometem a precisão, devido a limitações no seguimento dos utilizadores e à própria destreza humana.

Neste trabalho de investigação, procurámos reduzir o impacto da falta de precisão de gestos no ar em AV imersivos. Depois de analisar o estado-da-arte na manipulação de objetos em vários paradigmas de interação, começámos por avaliar o desempenho de técnicas existentes, tanto em AV semi e totalmente imersivos. Se não houver restrições, a melhor abordagem será usar mapeamentos exatos dentro de um AV totalmente imersivo. Focados em aumentar a precisão da manipulação de objetos no ar, investigámos se esta beneficia da separação dos graus de liberdade (GDL), que foi provada útil noutros paradigmas de interação. Os nossos resultados mostraram que o controlo de um único GDL pode melhorar a precisão, sacrificando tempo para tarefas complexas. Para testar se há vantagens em combinar separação de GDL com translações escaladas, e se eixos de transformação personalizados permitem desempenhos mais rápidos do que eixos fixos, desenvolvemos duas novas técnicas. Descobrimos que não há melhorias significativas em escalar translações isoladas, e que os utilizadores preferiram manipulações com 3-GDL, mantendo a translação e a rotação independentes. A precisão dos gestos no ar também tem um papel importante na seleção de objetos fora de alcance. Desenvolvemos uma nova técnica de seleção para AV imersivos, que combina um cone regulável com uma estratégia de refinamento progressivo iterativo, teleportando os utilizadores para mais perto dos objetos intercetados. Os resultados de uma avaliação de usuários revelaram a nossa técnica como sendo versátil para a aquisição de alvos distantes, combinando seleção precisa com tempos consistentes em diferentes cenários.

Em conclusão, validámos a nossa tese, que afirma que a separação de graus-de-liberdade e estratégias de refinamento progressivo iterativo podem ser usadas para oferecer interações no ar em ambientes virtuais imersivos mais eficazes.

Keywords

3D User Interfaces
Precise Mid-air Object Manipulation
Immersive Virtual Environments
DOF Separation
Iterative Progressive Refinement

Palavras Chave

Interfaces de Utilizador 3D
Manipulação Precisa de Objetos no Ar
Ambientes Virtuais Imersivos
Separação de Graus de Liberdade
Refinamento Progressivo Iterativo

Acknowledgements

During the execution of the work presented in this document, it was the support, collaboration and encouragement of many people that allowed me to achieve this goal, to whom I owe my sincere acknowledgments. First of all, to my supervisors, Professor Alfredo Ferreira and Professor Joaquim Jorge, for all their continuous guidance, advices and valuable insights along these years.

In second place, to all members of the VIMMI group at INESC-ID for their aid, specially to those that shared an office with me: Bruno Araújo, Maurício Sousa and Daniel Pires, and also to Rafael Kuffner. Your availability to discuss ideas and contributions made during the most critical deadlines were invaluable. To Professor Andrea Giachetti and Fabio Marco Caputo, from the University of Verona, for their valued contributions to the literature review and respective discussion. To all the Master's students that accepted the challenge of collaborating with me, helping in the development of prototypes and conducting user evaluation sessions, namely: Fernando Fonseca, Vasco Rodrigues, Filipe Relvas, Eduardo Cordeiro and Rodrigo Lorena. Additionally, to all the people who found the time to participate in all the user studies, without asking for anything in return.

To the institutions that allowed me to perform the work presented in this document through financial support: Fundação para a Ciência e a Tecnologia (FCT) through the doctoral grant SFRH/BD/91372/2012, and research projects Alberti Digital (PTDC/AUR-AQI/108274/2008), CEDAR (PTDC/EIA-EIA/116070/2009), TECTON 3D (PTDC/EEI-SII/3154/2012) and IT-MEDEX (PTDC/EEISII/6038/2014); and my host institution INESC-ID Lisboa, under contract UID/CEC/50021/2013.

Last but not least, to my family for their unconditional love and support. In particular, to my parents, Alice and José, who allowed me to be who and where I am today, and to my wife, Susana, who dealt gracefully with all the stress and the emotional roller-coaster during this journey.

Many thanks to all of you!

Para a minha filha Maria Beatriz

Contents

List of Figures

List of Tables

1 Introduction

Over the last decades, computers have became ubiquitous in our society. From super powerful workstations to highly portable tablets and smartphones, the use of computer systems is commonplace in almost every field. Among others, architecture and engineering are remarkable examples, where these systems provide critical aid in design and plan stages. They allow for highly accurate technical drawings, schemes and models, and ease the corresponding review by allowing a process free of physical artifacts. This both speeds up processes and reduce costs.

When computer systems are used to support this kind of work, tasks take place on virtual environments (VEs), synthetic worlds where the view upon them is under the real time control of the user [73], instead of occurring in the physical world. Traditionally, the visualization of the VE is done with screens that display a simple projection of the environment in a two-dimensional (2D) plane. Although this suits 2D VEs, the perception of three-dimensional (3D) VEs can be enhanced by combining stereoscopic (or stereo) visualization with head tracking, to increase user immersion through additional depth cues.

A stereo visualization can be achieved using stereoscopic screens or with head-mounted displays (HMDs). Stereoscopic screens, such as the nowadays common 3D televisions (3DTVs) or projectors, extend traditional screens by creating the illusion that the virtual content is not restricted to the screen plane. This can be

used, for instance, to show virtual objects in mid-air, creating a semi-immersive virtual environment (SIVE). If a user is completely surrounded by these kind of screens, typically in CAVE[1] settings, his reality can be replaced by a totally virtual one - a virtual reality (VR). However, this kind of setups are very expensive. Alternatively, HMDs can also be used to create immersive virtual environments (IVEs) by providing a different screen to each eye and totally occluding the physical space surrounding the user.

While HMDs are not new, recent technological advances made hardware better, affordable and widely available, which led to a regained interest in VR. Today, IVEs made possible with these kinds of technologies are being used for several purposes - not only engineering and architecture, but also medicine, manufacturing, game development, entertainment, and so forth - offering unique capabilities.

Other recent technological advances have also made it easier to develop interactive IVEs. Not long ago, user tracking required expensive and invasive systems. Currently, user tracking is possible using affordable and non-intrusive methods based on depth cameras, infra-red cameras and markers on headsets and controllers, and low-latency inertial sensors. These tracking solutions can be used not only to find the user's point-of-view, but also to track user limbs and hands, unveiling new interaction possibilities. The combination of stereoscopic displays and user tracking allows users to naturally manipulate 3D entities as if they were co-located with their hands and body, extending traditional 2D interactions in very natural ways.

1.1. Motivation

To interact within VEs, the ability to manipulate virtual objects is a key feature. Being of such importance, the search for effective methods for selecting, translating and rotating virtual objects has been a major research target. These objects can be any virtual entity with modifiable location and/or orientation properties. A virtual object can be, for example: anything ranging from a column to an entire building in architecture; an immense oil platform or a single bolt in engineering projects; an implant or an atom within a molecule in bio-medical scenarios; or even a vertex in virtual modelling which can be used to create entirely new content.

Considering objects in 3D VEs, manipulations are not trivial mainly due to the required mapping between traditional input devices and the VE. Most of the common

[1]CAVE: Cave Automatic Virtual Environment.

solutions resort to techniques that somehow relate the actions performed in the 2D space of the input device (e.g. mouse cursor or touch) to 3D transformations. Aiming to offer more natural interfaces, touch-enabled surfaces introduced the possibility of directly interacting with virtual content. Although having a 2D input similar to mouse-based interfaces, users are able to touch the virtual objects that they want to manipulate. Additionally, by allowing simultaneous touches, interfaces can have a higher input bandwidth, leading to new manipulation techniques.

To overcome the limitations of both the input and the output devices, mainstream solutions for creating and editing 3D virtual content, namely, computer-aided design (CAD) tools, resort to different orthogonal views of the environment. This allows a more direct 2D interaction with limited degrees-of-freedom (DOF). Other solutions offer a single perspective view, and generally apply transformations either in a plane parallel to the view plane, or resort to widgets that constrain interactions and ease the 2D-3D mapping. Research has shown that the first approach can occasionally result in unexpected transformations when users are allowed to freely navigate through the VE and that constrained interactions allow for more accurate manipulations.

Due to these workarounds to deal with non-trivial input/output mappings, traditional software tools for CAD and 3D manipulation have very steep learning curves. As such, these are mostly used by expert professionals with specific training and a lot of experience, being highly difficult to use for novices users who want to create or edit 3D virtual content.

Alternatively, spatial tracking solutions make it possible to know where users' heads, limbs and hands are in 3D. Combining these with immersive visualization devices, which hinder the use of traditional input devices since they occlude users' physical reality, can be used to enable virtual manipulations mimicking interactions with physical objects. This allows for more direct and natural interactions in mid-air over traditional 2D interfaces, being easier to learn and use.

1.2. Challenges

Although mid-air manipulations show promising results due to their naturality, there are still several open challenges regarding interactions with virtual content in mid-air. The most popular approach for 3D object manipulations within IVEs consists in simulating those with physical objects: grab and directly move and rotate with

simultaneous 6 DOF. However, it is still difficult to place a virtual object in the desired place with a high level of accuracy. These difficulties may arise from different factors, such as limited human dexterity for mid-air gestures and lack of precision from tracking systems. Although an accurate object placement is not required in all applications (e.g. visualization), precision is of extreme importance when creating or assembling engineering models or architectural mock-ups, for instance. While with traditional WIMP[2]-based interfaces it is possible to specify exact values to objects' transformations, mid-air approaches are still only suitable for coarse actions.

Due to the aforementioned limitations, precise manipulations of virtual objects are mainly performed in desktop setups, with techniques still very similar to those proposed more than 20 years ago. Despite the many advances in VR we have been witnessing in that period, IVEs are almost solely used for either entertainment or pure visualization purposes. For example, when engaged in engineering projects, users typically resort to immersive setups to have a better understanding of the virtual content, take notes of what needs to be modified, and then perform the intended modifications back in the desktop computer with CAD tools. This presents us the first challenge we aim to tackle in this thesis:

> *How can placement precision of mid-air spatial manipulations be improved, when using immersive virtual environments?*

Additionally, when interacting in VEs, object identification is essential so that the system can understand to which virtual object should manipulations be applied to. Indeed, every action performed on a virtual object can be decomposed into three stages: selection, manipulation and release [21]. Object selection is traditionally done in WIMP interfaces by placing the cursor on top of the object and clicking, or, more recently, by directly touching it on an interactive surface. In VEs that support mid-air hand tracking, object selection is usually performed by directly grabbing the desired object. Therefore, objects placed outside users' reach are also attention worthy.

To overcome this physical constraint, techniques that follow an arm-extension metaphor have been proposed to allow users to select out-of-reach objects. A different approach that requires less physical movement, consists in a natural pointing metaphor. However, for both these approaches, the more distant the object is, the lesser accuracy users have, since a small hand tremor or tracker jitter can drastically move the ray or arm away from the desired object. This leads us to our second challenge:

[2]WIMP: windows, icons, menus and pointing devices.

When in virtual reality settings, how can users effectively select virtual objects that are outside their arms' reach?

1.3. Thesis Statement

IVEs are very appealing for several fields, such as engineering and architecture, among others. In this context, the capability to select and precisely move and rotate 3D virtual elements assumes a great role. Mid-air manipulations, often supported by 6 DOF spatial tracking, allow for very natural interactions. However, strictly natural techniques have not only advantages, but also disadvantages. While they are easy to learn and use, they possess the restrictions people have in the physical world. Practical examples are the limited human dexterity in mid-air, and the interaction with objects outside arm's reach which is not physically possible. Even the act of pointing to select distant objects is hampered by problems of dexterity and accuracy. These are the challenges we want to address. Thus, we present the thesis statement of this dissertation:

> **Precision of mid-air manipulations in immersive virtual environments, including selections and transformations, can be enhanced over direct and natural approaches by exploiting hyper-natural interactions, such as the separation of degrees-of-freedom and iterative progressive refinement strategies.**

1.4. Context

Along this thesis, I have been involved in four national research projects (Alberti Digital, CEDAR, TECTON 3D, and IT-MEDEX). All these projects addressed several challenges related to 3D user interfaces applied in different fields. Alberti Digital and TECTON 3D were focused on architecture. In the first, we developed an interactive setup to allow public expositions' visitors to explore Alberti's architectural treatise of the Renaissance. TECTON 3D proposed new techniques for 3D interaction based on hand gestures, enabling mid-air 3D modelling to create architectural mock-ups. The CEDAR project targeted collaborative review of 3D virtual models related to engineering projects, namely in the oil industry. Lastly, in IT-MEDEX

we conducted research on novel ways to visualize and interact with 3D medical content.

Collaborating with these projects enabled us to experiment several technologies and tools, and to work with professionals from distinct areas. In all these projects, the interaction with 3D virtual content in immersive setups was a constant. Taking advantage of the most recent technological advances, we developed several novel interaction techniques, exploring distinct metaphors with different input modalities, namely mid-air gestures. This helped us identifying the challenges pointed, and gave the context for several of the user experiments described in this dissertation.

1.5. Approach and Hypotheses

Instead of narrowing to natural interactions, we explored magical interactions or hyper-naturalism, which "allows users to make use of their existing knowledge and understanding of the world but is not limited in the same ways that the real world is" [73]. During this work we also followed other tips from previous research, such as using principles from 2D interaction to design 3D techniques and reducing DOF whenever possible [73].

Being part of the Human-Computer Interactions field, 3D user interfaces are usually validated through user evaluations. As such, all the stages in this thesis involved user evaluations, both in initial assessments and to validate our hypotheses.

Given its relevance, 3D object manipulation in VEs has been subject of research for long, covering different kinds of interaction paradigms. Our first step was to identify which existing mid-air techniques for 3D virtual object manipulation perform best and appeal the most to users. Then we studied whereas those identified techniques are suited both for SIVEs and IVEs, since not all were conceived for the same environments. These assessments allowed us to set the baselines for the following evaluations.

Next we tackled our first challenge, focused in enhancing the precision that can be attained with mid-air manipulations. While aids such as snapping to grids or other objects make manipulations faster and more precise, they also restrict freedom of input, which is a requirement in creative tasks. Therefore, our objective was to increase user precision in mid-air while allowing every possible location and orientation for the manipulated object.

A path that can be followed to attain more precise and controlled 3D object manipulations is to follow DOF separation. DOF separation first appeared to overcome the mapping difficulties between 2D input and desired 3D output, first for mouse-based interfaces and then for touch interactions. It led to better users' performance when compared to direct approaches, and it has been shown that controlling only 1 DOF at a time can be beneficial.

While in mid-air interactions this dimensional difference between input and output does not exist, we reckon that it can also benefit from highly controlled manipulations, since mid-air input has even more DOFs. Indeed, previous work showed that even in mid-air, users tend to decompose complex tasks into sequences of single DOF manipulations. Although this reduces the naturalness of direct 6 DOF interactions, with an hyper-natural approach we might extend human capabilities in ways that are not possible in the physical world, which can be advantageous in certain scenarios.

A common way to achieve DOF separation is through virtual widgets, which allow users to select specific transformations and axes. Virtual widgets for 3D manipulation became a *de facto* standard in mouse-based 3D user interfaces, are becoming ever more common in multi-touch applications, and have even been proposed for mid-air interactions. While for the first two results suggest that this approach is successful, in mid-air there are still challenges to be tackled. Although using widgets themselves is not a natural approach, interacting with them can be. Grabbing and moving an handle, for instance, can be performed with a direct approach, while restricting its movement using the metaphor of it being on rails.

While for pure performance purposes close mapping of input and output DOFs is desirable, as stated by several researchers, this is not true when more accurate positioning is in order. We resorted to virtual widgets in mid-air to restrict manipulations to a single DOF to test our first hypothesis:
Hypothesis 1 *Mid-air manipulations with single DOF control reduce placement error in docking tasks over direct approaches with an exact mapping.*

Although 1 DOF manipulations might reduce errors to a single dimension, they do not totally overcome the limited accuracy in mid-air. This challenge has been subject of previous research, but a definitive solution is yet to be found. It has been suggested that approaches that enhance precision, such as scaled hand motions, can be an improvement over direct mappings. We believe that manipulations with DOF separation also benefit from such approaches, which lead to our second hypothesis:

Hypothesis 2 *DOF separation in mid-air can be combined with scaled user input to increase precision in 3D manipulation tasks.*

When restricting manipulated DOFs using widget based approaches, users can select an axis to apply a transformation. This axis is usually from the word or object frames. Hence, transformations along more than one axis require multiple operations, which lead to additional task completion times. Some techniques developed for 2D input that follow DOF separation without widgets allow users to specify custom transformation axis. We think that this can contribute for faster and more straightforward manipulations. As such, our third hypothesis can be put as follows:

Hypothesis 3 *The possibility to specify custom transformation axes leads to faster mid-air manipulations than using object or world axes, while keeping the same level of precision.*

Finally, we addressed our second challenge, regarding the selection of out-of-reach objects in IVEs. As with mid-air manipulations, pointing to an object can be severely hindered by tremors or tracker noise. To reduce this effect, the ray used to point can be exchanged by a cone, increasing selection space. Nevertheless, this method still has drawbacks: if the aperture of the cone is too small the same problem of the ray arises, and if it is too big several objects will be intersected, requiring disambiguation techniques. This later issue can be overcame by using a Progressive Refinement strategy, which has been firstly proposed to interact with traditional displays, and has been scarcely explored in IVEs.

Existing Progressive Refinement techniques often favor closer objects either rearranging them or automatically refining selection based on proximity, which might not be feasible for selecting far-away objects. Other techniques employ menus or zoom metaphors, and were developed for non-immersive and non-stereoscopic scenarios. Thus, they may not be suited for IVEs since such kind of object rearrangement might disrupt user immersion, and zoom approaches can lead to cybersickness. Nonetheless, it is our belief that a variation of these techniques can be effectively used in IVEs, moving the user closer to selected objects instead of modifying the field-of-view to zoom in. This can be repeated until the desired object can be selected with ease. Thus, the fourth hypothesis we evaluated is:

Hypothesis 4 *A progressive refinement strategy, using cone-casting and iteratively moving users closer to intersected objects, can be used in IVEs as a mean for accurate selections outside arms' reach.*

1.6. Contributions

The research we conducted in this thesis led to the following scientific contributions in the field of Human-Computer Interaction, 3D User Interfaces and Virtual Reality:

- Assessment of the most efficient and preferred techniques for virtual object manipulation in mid-air. We compared five techniques based on existing literature, with a set of docking tasks requiring translation, rotation and scale, and analyzed completion time and user preferences. Additionally, we also evaluated the performance of the overall best two techniques in both semi and totally immersive settings. The attained results, presented in Chapter 3, can help developers choose the technique that is best suited for their scenario.

- A study on the impact of explicit DOF separation in mid-air manipulation tasks, namely on time and precision. In this study, detailed in Chapter 4, we evaluated three manipulation approaches based on existing literature: one follows a direct metaphor, the second scales users' movement and the third is our implementation of mid-air virtual handles for DOF separation. From the results, we could draw a set of guidelines that should aid researchers and interaction designers to create techniques that can take advantage of the better aspects of each evaluated approach.

- Two new techniques for mid-air object manipulation that explore DOF separation and scaled user movements. These techniques, described in Chapter 4, implement the previous guidelines in different manners, supporting translation, rotation and scaling, and allowing the simultaneous control of different combinations of 1, 2, 3 and 6 DOF. User evaluations comparing these techniques against previous ones highlighted their advantages and disadvantages, leading to further insights useful for future mid-air manipulation techniques.

- A novel selection technique for objects outside users' arms reach in IVEs. It is the first to employ an iterative progressive refinement in such settings, as presented in Chapter 5. With this technique, users are able to select groups of objects using a cone-casting approach, being instantaneously moved closer to them. This process can be iteratively repeated, until the desired object may be easily selected. A user evaluation showed that our technique is a versatile approach, combining accurate selections with consistent task completion times across different scenarios.

- Several setups to interact in mid-air with virtual content. To evaluate our hypothesis and proposed techniques, we developed multiple prototypes with innovative setups resorting to state-of-the-art technologies. We focused on non-invasive user tracking solutions, mostly based on depth cameras, combining them with other technologies such as accelerometers, gyroscopes and pressure sensors. User input was co-located with stereoscopic visualization devices. For this, we explored both 3DTVs (Chapter 3) and HMDs (Chapters 3, 4 and 5).

1.7. Publications

The work developed during this thesis yielded more than 30 publications accepted in peer-reviewed journals, conferences and scientific meetings. From those, we highlight the following, listed in chronological order by date of publication from the newest to the oldest:

1. **Daniel Mendes**, Fabio Marco Caputo, Andrea Giachetti, Alfredo Ferreira, and Joaquim Jorge (2018). *A Survey on 3D Virtual Object Manipulation: from the Desktop to Immersive Virtual Environments.* Computer Graphics Forum. DOI: 10.1111/cgf.13390
   Main topic: Survey on the state-of-the-art in 3D object manipulation, ranging from traditional desktop approaches to touch and mid-air interfaces, to interact in diverse virtual environments, as presented in Chapter 2.

2. **Daniel Mendes**, Maurício Sousa, Rodrigo Lorena, Alfredo Ferreira, and Joaquim Jorge (2017). *Using custom transformation axes for mid-air manipulation of 3D virtual objects.* In Proceedings of the 23rd ACM Conference on Virtual Reality Software and Technology (VRST '17), pp. 27:1–27:8. DOI: 10.1145/3139131.3139157
   Main topic: Evaluation of a novel manipulation technique that allows users to specify custom transformations axis for mid-air manipulations, presented in Chapter 4.

3. Maurício Sousa, **Daniel Mendes**, Rafael Kuffner Dos Anjos, Daniel Medeiros, Alfredo Ferreira, Alberto Raposo, João Madeiras Pereira, and Joaquim Jorge (2017). *Creepy Tracker Toolkit for Context-aware Interfaces.* In Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17). pp. 191-200. DOI: 10.1145/3132272.3134113

Main topic: Spatial user tracker with multiple depth cameras, described in Appendix A, and used in prototypes presented in Chapters 4 and 5.

4. **Daniel Mendes**, Daniel Medeiros, Maurício Sousa, Eduardo Cordeiro, Alfredo Ferreira and Joaquim Jorge (2017). *Design and evaluation of a novel out-of-reach selection technique for VR using iterative refinement.* Computers & Graphics Volume 67, pp. 95-102. DOI: 10.1016/j.cag.2017.06.003
   Main topic: New selection technique for out-of-reach virtual objects in IVEs and its evaluation, presented in Chapter 5.

5. **Daniel Mendes**, Filipe Relvas, Alfredo Ferreira, and Joaquim Jorge (2016). *The benefits of DOF separation in mid-air 3D object manipulation.* In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (VRST '16), pp. 261-268. DOI: 10.1145/2993369.2993396
   Main topic: Assessment of the impact of DOF separation in mid-air virtual object manipulations, as presented in Chapter 4.

6. Daniel Medeiros, Eduardo Cordeiro, **Daniel Mendes**, Maurício Sousa, Alberto Raposo, Alfredo Ferreira, and Joaquim Jorge (2016). Effects of speed and transitions on target-based travel techniques. In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (VRST '16), pp. 327-328. DOI: 10.1145/2993369.2996348
   Main topic: Evaluation of three travel techniques, described in Appendix B, to select the best suited to be used in the technique for out-of-reach selections presented in Publication 4 and Chapter 5.

7. **Daniel Mendes**, Fernando Fonseca, Bruno Araújo, Alfredo Ferreira, and Joaquim Jorge (2014). *Mid-air interactions above stereoscopic interactive tables.* IEEE Symposium on 3D User Interfaces (3DUI '14), pp. 3-10. DOI: 10.1109/3DUI.2014.6798833
   Main topic: Assessment of the most natural and efficient techniques from the literature for manipulating mid-air virtual objects, presented in Chapter 3.

1.8. Dissertation Outline

This dissertation is organized in six chapters. Following this introductory chapter, the outline of the document is structured as following.

In Chapter 2, we start by explaining concepts relevant to properly understand and discuss manipulation of virtual objects. Then, we present a survey on the state-of-the-art regarding virtual object manipulation with different interaction paradigms, covering traditional WIMP interfaces, touch-based interactions and mid-air approaches. In its discussion, we compare the presented works and highlight trends, considerations and open challenges. We also present exploratory work we did during this thesis that motivated our approach. Chapter 3 describes our initial user evaluations to identify the most natural and preferred mid-air manipulation techniques. We first evaluate a set of mid-air techniques against each other and against a multi-touch baseline, in a SIVE. Using the two techniques that performed better, we also assess how they perform in an IVE versus a SIVE.

Chapter 4 details our research on methods to increase the precision of mid-air manipulations. We implemented mid-air virtual widgets to restrict transformations to a single axis from the object's frame, and compared it against a direct approach and an existing technique that scales down users' motion input. From this evaluation, we propose a set of mid-air manipulation guidelines, which we implemented in two novel techniques: WISDOM and MAiOR. The latter also allows users to specify arbitrary transformation axis, in an attempt to speed up manipulations while preventing errors. Both techniques were subjected to separated user evaluations.

In Chapter 5 we address the challenge of effectively selecting objects outside arm's reach. We introduce a new technique, PRECIOUS, which is the first to employ an iterative progressive refinement in immersive settings. It combines a manipulable cone as selection volume, and moves users closer to the intersected objects in each step, to ease selections. We also describe a user evaluation of PRECIOUS against two state-of-the-art approaches, and discuss the attained results.

Lastly, in Chapter 6 we present our conclusions to this thesis. We discuss the main results of the research conducted, and point out possible directions for future work regarding manipulation of virtual objects in mid-air.

2
Background and Related Work

VEs have been around for some time, and they are used for a myriad of purposes. Examples are bioengineering and geology [124], oil and gas [50], automotive engineering [92], manufacturing [94], architectural mockup [3] and CAD [61] to creative painting [64], animation movies [90] and entertainment [88]. VEs are nowadays ubiquitous in a plethora of fields.

The visualization of VEs and the interaction with them can be made in several manners: ranging from desktop computers equipped with traditional monitors, mouse and keyboard, to immersive settings such as the now increasingly common VR devices, and passing through multi-touch tabletops and surfaces, different hardware setups have been proposed. In these, being able to interact with virtual objects, especially manipulating them, is a major requirement.

In this chapter we explain, in Section 2.1, a set of fundamental concepts for object manipulation within virtual environments, introducing new taxonomies for 3D object manipulation that are useful for discussing existing techniques. In Section 2.2, we survey the most relevant research works regarding 3D virtual object manipulation for different environments, and discuss them accordingly. We then present, in

Section 2.3, exploratory work we developed during this thesis, which ultimately led us to the focus of this document.

2.1. Key Concepts

As VEs can be perceived in various ways, and also interacted with different modalities, we will start by overviewing the most common forms of input and output used. Next, focusing on object manipulation in said environments, we explain relevant concepts such as selection, transformations, degrees-of-freedom and input mapping.

2.1.1. Overview of Virtual Environments' Inputs and Outputs

User immersion in VEs can be enhanced by combining stereoscopic visualization and head tracking. By knowing the user's head position, it is possible to generate a visualization frustum to each eye to create the illusion of virtual objects being part of the physical world. This illusion is even stronger when users are allowed to freely move their heads and see different sides of a virtual object in their own perspective, without the need to manipulate cameras or widgets. Although HMDs and CAVEs (cave automatic virtual environments), which allow a fully immersive viewing experience, have existed for a while, interest in these technologies has increased considerably over the past few years. One of the main issues with older HMDs was the nausea that they caused, commonly referred to as virtual reality sickness or cybersickness.

However, the new generation of low-cost HMD devices that have recently appeared have demonstrated that this issue can be effectively solved by using low-latency inertial devices and smart rendering solutions such as the time-warping technique [35].

Other recent technological advances have also made it easier to develop immersive visualization scenarios. Not long ago, user tracking required expensive and invasive systems. Currently, user tracking is possible using affordable and non-intrusive methods based on depth cameras, IR cameras and markers on headsets and low-latency inertial sensors. These tracking solutions can be used not only to find the

user point of view to render the virtual scene but also to track user limbs and hands, unveiling new interaction possibilities. Additionally, this combination of stereoscopic displays and user tracking allows users to naturally manipulate three-dimensional entities as if they were co-located with their hands and body, extending traditional two-dimensional interactions in very natural ways.

A VE that can be explored through immersive displays is often called an immersive virtual environment (IVE). Although a fully immersive environment should explore other human senses in addition to vision, as studied by Azevedo et al. [7], the IVE classification is often used when using only an immersive display. According to Bowman et al. [22], these types of displays can be divided into two categories: fully immersive displays, such as HMDs, which completely occlude the real world, and semi-immersive displays, such as stereo tabletops, which allow users to see both the physical and virtual worlds. A VE that uses a semi-immersive display is commonly labeled as a semi-immersive VE (SIVE), and one that is perceived through a fully-immersive display can be said to be a virtual reality (VR). The benefits of higher levels of immersion have already been presented [19].

To describe the most relevant aspects of the VEs presented in the surveyed literature, we classify their properties following the organization proposed by Grossman and Wigdor in their analysis of tabletop interactions [49], adapted to generic environments.

Starting with the display properties, we distinguish conventional 2D displays from those providing stereoscopic depth cues regarding the space where imagery appears to be. We also differentiate this space from the actual space where the rendered images are presented. This is constrained to 2D for most of the current interaction setups because truly volumetric displays that illuminate voxels in mid-air are not common. When 3D perceived space is generated on 2D screens with stereo and motion parallax cues, issues such as hand occlusions may arise. To overcome this issue, there are heads-up surfaces, such as HMDs or see-through screens placed between the user's eyes and hands.

Another important characteristic of the rendering setup is the viewpoint correlation, which concerns the relation between the user's point of view and the viewpoint of the virtual scene. In systems where the user moves around the display and the viewpoint remains constant, there is no relation. For systems that change the viewpoint of the rendering according to the user's head position, we say that there is a high or total correlation. High refers to setups composed of a screen, either vertical or horizontal, that when the user moves his head behind the screen, he will see the back of the

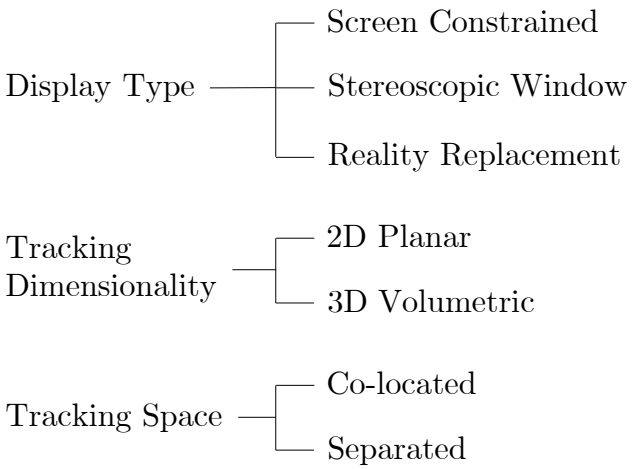


Figure 2.1: Overview of virtual environments' input and output properties.

screen rather than the VE from a different perspective. When using a HMD to create a virtual reality experience, total correlation between the user's viewpoint and the displayed imagery can be achieved.

Since there are several aspects that generally go hand-in-hand, according to different setups, we summarize some properties of virtual environments' inputs and outputs in Figure 2.1. Focusing on display type, VE visualization can be screen constrained, made through a stereoscopic window or be perceived as a reality replacement. Screen-constrained visualizations, such as those of traditional desktop displays, are based on rendering on 2D screens with no stereo depth cues and have no viewpoint correlation. Stereoscopic window, although also constrained to a 2D screen, offers a view of the VE with stereoscopic depth cues and high viewpoint correlation. With this visualization, virtual objects can appear to be within the screen (positive parallax), generally referred to using a fish tank metaphor; at the screen plane (zero parallax); or between the user and the screen (negative parallax). Using heads-up surfaces, fully immersive displays have total viewpoint correlation and employ stereo depth cues, replacing users' reality with the virtual one.

In these VEs, user interaction is often leveraged by tracking handheld devices or human body parts in 2D (e.g. mouse, touchscreens) or in 3D (through inertial or vision-based trackers). The tracking system may also allow the co-registration of the visualized and input spaces, allowing direct interactions with the virtual content. Despite acknowledging that there are several multi-modal interaction techniques that resort, for instance, to speech and/or gaze in addition to the aforementioned gestural input (e.g. [17, 113]), we will focus mostly on hand-based techniques.

These techniques can be either hands-free through multi-touch and mid-air input or through handheld devices, such as mouse or spatially tracked controllers.

As a last note, interface design should be carefully adapted to the type of immersive display used. There are differences between distinct reality replacement setups, such as HMDs and CAVEs, and stereoscopic windows. A number of interface issues arise with stereo displays, as stated by Bowman et al. [22]. For instance, because users can see their own hands in front of the display, they can inadvertently block out virtual objects that should appear to be closer than their hands. With HMDs, since users do not see the position and orientation of their bodies and limbs, solutions must be explored to increase users' proprioception [91].

2.1.2. Manipulation: Selecting Objects

We are interested in a particular type of interaction in VEs: the manipulation of existing virtual objects in the scene. To perform such manipulation, the system needs first to know to which object should users' actions be applied to. As such, and according to Bowman and Hodges [21], performing an object selection is the first step to execute a manipulation action.

To classify existing selection techniques, we propose a taxonomy, illustrated in Figure 2.2, which complements those proposed by Bowman and Hodges [21], Poupyrev and Ichikawa [104] and Kopper et al. [70]. The first [21] classifies the feedback given, how to indicate the object and how to confirm the selection. The second [104] classifies techniques according to interaction metaphors, while the latter [70] classifies the progressive refinement strategy. Our taxonomy focuses instead on cardinality, i.e. the number of objects selected, reach and offers a new nomenclature to the strategy followed for the progressive refinement.

Techniques that can only select one object are classified as having Single Cardinality. On the other hand, those that are capable of selecting multiple objects, are referred to as having Multiple Cardinality. The latter can be further decomposed into Serial and Parallel [77], depending on whether the several objects are selected one at a time or at the same time, respectively.

Reach represents how far from the user the object selection can be done. Techniques where the environment is displayed in a traditional non-stereo screen are classified as Screen-Space. These approaches allow users to select a point on the screen and the object portrayed in such position will be selected, independently of how far it

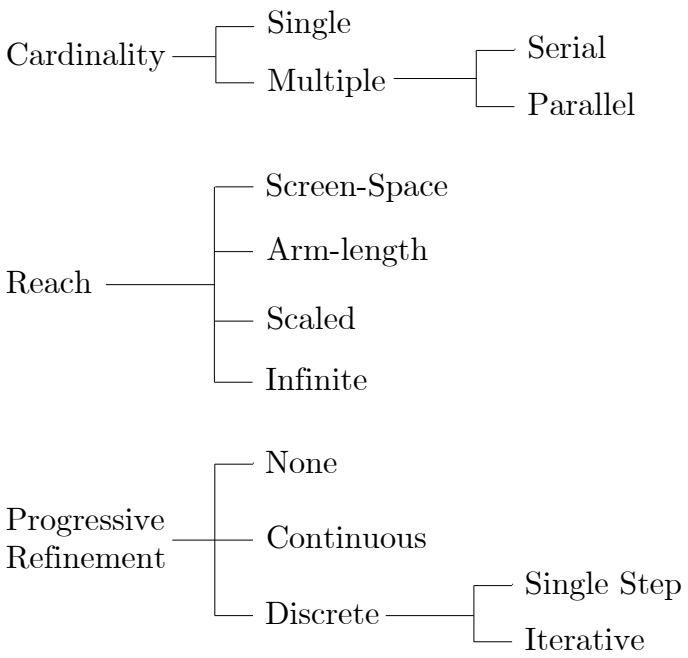


Figure 2.2: Taxonomy of object selection techniques' properties.

is. Arm-length refers to techniques where the length of the user's arm limits where the user can reach objects for selection. Techniques that are classified as Scaled are the ones where the extent of users' reach is greater than its arms length, but can not reach infinite. Scaling effect is usually achieved by using an anisomorphic control-display ratio [4]. Infinite reach classify techniques where there is no limit to where objects can be so that users are able to select them. A common example of this is the ray-casting technique where the selection of objects are made through the intersection with the object and a mathematical ray.

In some cases, the selection can be imprecise and lead to an undesired selection of multiple objects. This can be overcame by using a Progressive Refinement strategy in the selection process. Progressive refinement was first introduced by Kopper et al. [70] and refers to the process of gradually reducing the amount of selectable objects among a larger group. This strategy is divided in two phases: the first, which defines the group of interest and the second, which consists in disambiguating between the objects within the group by selecting fewer objects. The progressive refinement strategy can be done by a continuous process (Continuous) or by one (Discrete, Single Step) or many discrete refinement operations (Discrete, Iterative).

2.1.3. Manipulation: Transformations and Degrees-of-Freedom

Manipulation is the task of changing the characteristics of a selected object [21]. For instance, it can be considered as an application of spatial transformations; change of visual properties, such as colour or texture; or even free-form deformation. However, manipulations most commonly refer to spatial transformations [21, 73].

Several different types of spatial transformations exist: translation, rotation, scaling, shearing, and reflection, among others. Although there are research works that cover all these transformations in VEs, the most common transformations are translation and rotation, which are required for positioning tasks. These transformations are also the ones that have the greatest resemblance to everyday physical interactions. Nonetheless, since the seminal works [98, 136], scaling has been grouped with these two basic operations. This trio of transformations, identified as the basic manipulation tasks [73] along with selection, has been kept together in a plethora of other research works; several are described in this document, whereas the remaining transformations are not considered. Moreover, these three transformations generally appear together in commercial 3D software, such as Blender and Unity3D.

Positioning manipulations can be performed in diverse ways, either on a single object in isolation disregarding its surroundings, or by aligning and snapping to other objects in the scene [15, 109], or even by grabbing multiple objects and aligning them, either packing or evenly distributing, or simultaneously moving them [116]. Ultimately, however, "any 3D manipulation can be constructed by translations and rotations around the object origin" [109]. Therefore, we will focus on the basic canonical manipulation tasks, namely translation, rotation and scale [73], of single virtual objects.

Each of these transformations can be applied to three different axes (x, y, z). A single transformation on one of these axes is commonly referred to as a degree-of-freedom (DOF). Thus, for a system that allows all transformations in all these axes, it is said that it allows transformations in 9 DOFs. For systems that only offer translation and rotation in 3D, they are said to support 6 DOFs, and for those that add to this uniform scaling, it is said that they support 7 DOFs. DOF is also used to specify devices' tracking capabilities. For example, a mouse can track position changes in a plane (2D); thus, it is a 2-DOF device. A spatial wand, whose position in space (3D), pitch, roll and yaw are tracked, is a 6-DOF device.

2.1.4. Mappings and Remappings of Transformations

Transformations are enabled by *mapping* a user's input onto actions performed on the manipulated object. This can be performed either through physics simulation or pre-programming specific interaction behaviour using, for example, gesture recognition. For the first, user input can be mapped to contact forces due to friction and collisions from virtual proxies within the physics simulation, enabling emergent hand-based gestures [134, 59]. These can include, for instance, sweeping, scooping, lifting, and throwing virtual objects [59]. However, "some aspects of traditional interactions do not naturally lend themselves to a physics implementation" [134]. For example, dynamically scaling an object cannot be implemented through a rigid-body simulation. Consequently, we will focus on pre-programmed interactions, where input DOFs are explicitly mapped onto manipulated object transformation DOFs. Input DOFs can be those derived by tracking position and orientation in 3D, but they can also be measurements of user actions obtained through other input channels (buttons, trackballs, and isotonic and elastic sensors [44]).

To classify the different mappings used in virtual object manipulation in VEs, we developed a taxonomy, presented in Figure 2.3, based on that proposed by Bowman and Hodges [21]. Whereas Bowman and Hodges covered all steps involved in a manipulation task, here we focus on transformations and go further in this component. Additionally, transformations can be applied simultaneously, having *no* separability, as it occurs with physical manipulations (translation and rotation) and common multi-touch interactions where users can move, rotate and scale objects with a single gesture. Transformations can also be applied separately, as is common in 3D modelling/editing software, requiring users to apply a single transformation at a time. However, some manipulation techniques group different transformations or only some DOFs from a transformation type, while separating the others. For these, we refer to them as having *partial* transformation separability. We only consider a technique to have *total* transformation separability when it enables users to perform every supported transformation in isolation from the others, e.g. it is possible to move an object to a new position along all axes without performing a single modification to its orientation or scale.

To map users' input onto transformations, several approaches can be followed. An *exact* manipulation maps the spatial transform of a device or a hand tracked directly onto the virtual object transform. In other words, it offers a 1:1 control, even if the tracked input and the virtual object have a fixed offset. If the tracked hand/device is co-located with the virtual one, then the effect is a simulation of a real-world

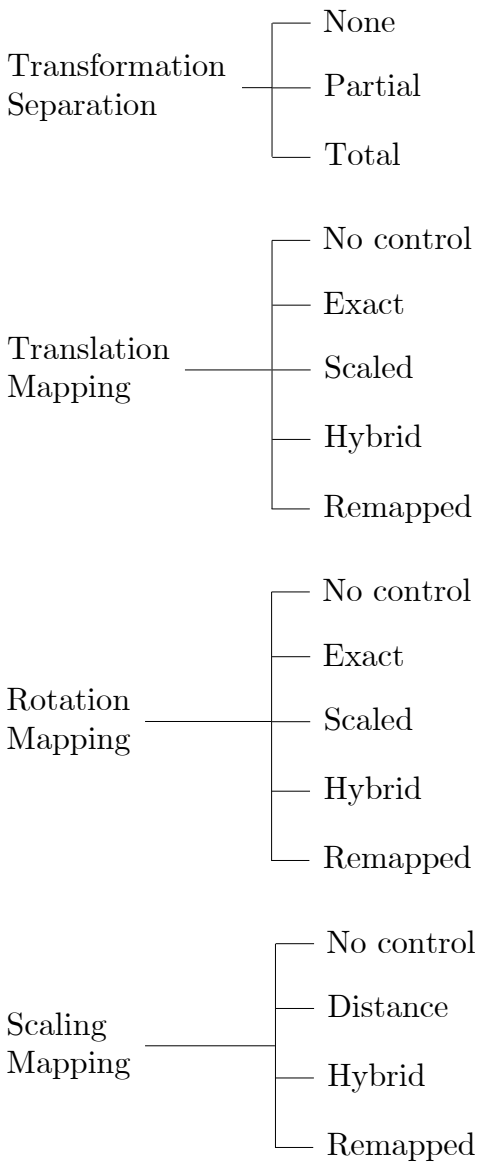


Figure 2.3: Taxonomy for classifying different approaches for object spatial transformations in virtual environments.

manipulation. The selected translational or orientation DOF of the tracked input can also be mapped directly onto the virtual world's ones or with a linear or non-linear *scaled* transform to increase accuracy or obtain increased ranges of transform parameters through N:1 or 1:N controls, respectively.

To overcome the limitations due to physical constraints, to allow 3D manipulation with 2D tracking, or to limit manipulated DOFs when having higher input DOFs, many techniques rely on indirect mappings. These mappings map tracked DOFs onto different manipulation DOFs (e.g. a slider controlling rotation, virtual widgets applying restrictions or specific gestures outside the object to trigger additional

transformations) or use different input channels to control object transform DOFs (e.g. mouse, keyboards, joysticks, microphones, and so forth). This *remapping* procedure might involve "learning a sensorimotor mapping that produces different results in a virtual world than one would expect from the real world" [78], and it is probably the most critical design issue in the development of a manipulation technique because it is difficult to find an optimal solution for different contexts. Mapping should allow the user to exploit existing or easy to learn motor programs [78], making the interaction effective, easy to learn and easy to use. This result is often searched through the use of metaphors.

However, there are techniques that apply different mappings to different DOFs of the same transformation, e.g. exact 1:1 control to a subset of the DOFs and remapping through widgets for the others. We define these transformation mappings as *hybrid*.

Direct manipulation with an exact mapping is not always the best solution, particularly when pursuing maximum accuracy or when we want to allow large translation, rotation and scaling. It is generally desired in immersive VEs, but it is perfectly acceptable, even in such environments, to remap different tracked motions or actions on buttons or joysticks onto object transforms. This is a typical solution in modern VR games, for example. We will discuss design choices in Section 2.2.4.

For scaling, since an exact mapping does not exist because this transformation is not physically possible, the most common is a *distance* mapping. This mapping resorts to the variation in the distance between two input points, using the metaphor of "stretching a piece of rubber" [136]. This mapping was suggested long ago [98] and made popular to the common public with the advent of touch-enabled mobile devices. However, different approaches for performing scaling transformation exist, which remap the input differently.

2.2. State-of-the-art

In this section, we introduce the vast amount of manipulation methods proposed in the literature that exploit 2D tracking, multi-DOF devices and 3D tracking techniques. We will first address traditional desktop interactions with screen-constrained visualization and mouse-based 2D input. Then, we cover touch-based manipulation with both screen-constrained visualization settings and stereoscopic tabletops. Lastly, we report on manipulations based on mid-air input. Since the first step in 3D

manipulation is object identification, we will also cover selection techniques. A thorough survey regarding that topic was made available by Argelaguet and Andujar [4], and we will refer the most relevant research for our work. The surveyed literature is followed by a discussion, where we identify trends and open challenges.

2.2.1. Desktop 3D Interfaces

Many computer applications, such as architectural modelling, virtual model exploration, engineering component design and assembly, require virtual three-dimensional object manipulations, among others. To work with VEs for this purpose, several interaction techniques for traditional desktop setups have been explored, resorting to mouse input.

2.2.1.1. Clicking to Select

The first step towards applying a transformation is to select an object. In mouse-based interfaces, this is done by placing the mouse cursor on top of the desired object's 2D projection on the screen and performing a click action. This approach can be used to select any object appearing on the screen, independently of its distance to the camera or user position. Thus, selections with mouse input have a screen-space reach. Due to the accuracy of mouse input, no refinement strategies are usually required. However, some of the progressive refinement approaches presented in Section 2.2.3.5, that were developed for non-immersive and non-stereoscopic large displays, can be adapted to mouse input.

2.2.1.2. Traditional Mouse-based Manipulations

To manipulate objects in this kind of interfaces, Nielsen and Olsen [98] created the triad cursor to overcome the mapping of 2D mouse input to 3D. Mapping is performed by comparing its screen-space movements with the projected image of its three perpendicular axes. By also taking advantage of the projections of the object's features, it allows separate translation, rotation and scaling transformations according to the object's edge or a plane defined by a face of the object. Zeleznik et al. [136] used two cursors, one controlled by each hand, to simultaneously perform the three different transformations restricted to a pre-specified plane in 3D.

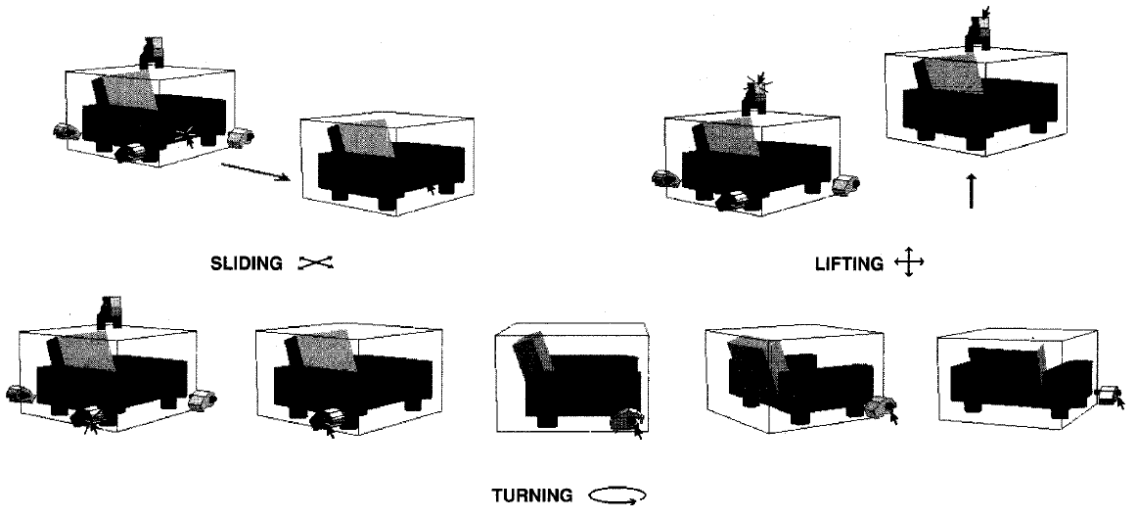


Figure 2.4: Sliding, lifting and turning a virtual object using the handle box approach (extracted from [60]).

Alternatively, Stephanie Houde developed an approach based on a handle box [60]. This approach consisted of a bounding box surrounding the object, and it had a lifting handle attached to it to move the object up and down and four rotation handles to rotate the object about its central axis, as illustrated in Figure 2.4. No handle was provided for sliding in the object's resting plane, on the assumption that the most direct way to slide an object would be to click and drag on the object inside the box itself. Conner et al. [32] also resorted to virtual handles to develop 3D widgets for performing transformations on virtual objects. They allow full 9-DOF control (translation, rotation and scaling) and even other deformations, such as twisting. The handles have a small sphere at their ends, and they are used to constrain geometric transformations to a single plane or axis (Figure 2.5). Dragging one of the spheres can translate, rotate or scale the object depending on which mouse button is pressed. For rotations, the direction of the user's initial gesture determines which of the two axes perpendicular to the handle is used as the rotation's axis.

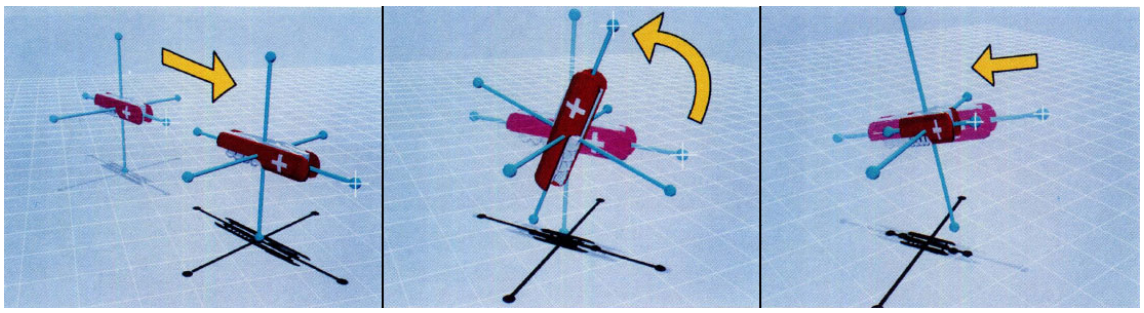


Figure 2.5: Virtual handles for object manipulation: translation (left), rotation (middle) and scaling (right) along a single axis (extracted from [32]).

Focusing only on rotations, Ken Shoemake proposed Arcball [111], an input technique that uses a mouse to adjust the spatial orientation of an object. To change the object's orientation, the user draws an arc on a screen projection of a sphere. For axis-constrained rotations, Arcball includes the view coordinate axes, the selected object's model space coordinate axes, world space coordinate axes, normals and edges of surfaces, and joint axes of articulated models (such as robot arms). Mouse, menu, or keyboard combinations can be used to select among axis sets. As an example, for body coordinate axes, three mutually perpendicular arcs would be drawn, which are tilted with the object. When the mouse is clicked down to initiate a rotation, the constraint axis selected will be that of the nearest arc.

More than 20 years have passed since these techniques were proposed, and they are still currently being used in several solutions, even commercial ones. Indeed, some applications that require object manipulation, such as Unity3D or SketchUp, resort to widgets both for mapping between input devices and corresponding 3D transformations and for restricting DOF manipulation. For interactively translating and scaling virtual objects, Unity3D, a commonly used game engine, allows users to do so through virtual handles, as depicted in Figure 2.6, similar to Conner et al. [32]. For rotations, it uses a direct implementation of Arcball [111]. SketchUp, a 3D modelling application, resorts to a handle box for object scaling, as also shown in Figure 2.6. It provides quick and accurate modelling, aided by dynamic snapping, input of exact values for distances, angles and radius. All these solutions allow users to perform a transformation in a single axis at a time.

Other commercial applications, namely, those for 3D modelling, often present a different option. Rather than using widgets to restrict DOF manipulation, they allow the 3D VE to be presented through three orthogonal views. Examples of this are 3D Studio Max or Blender (Figure 2.7). In this way, each view allows simple

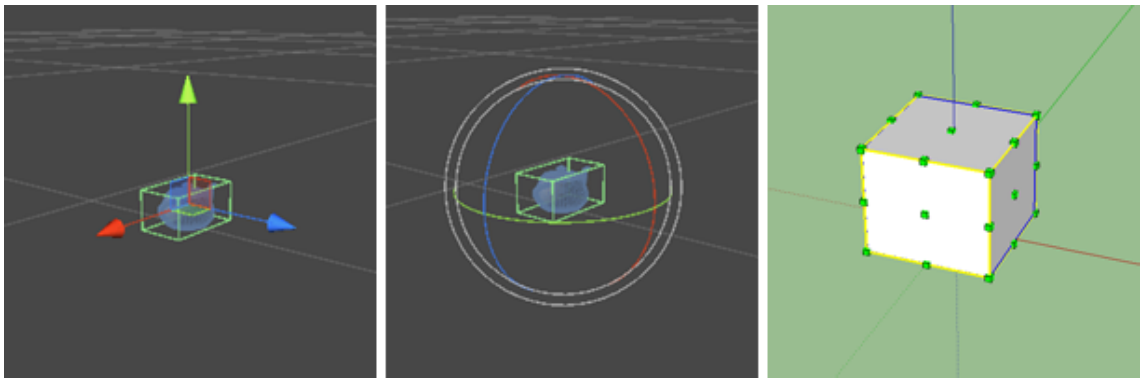


Figure 2.6: Widgets used in current commercial applications: virtual handles (left) and Arcball (middle) in Unity3D; handle box (right) in SketchUp.

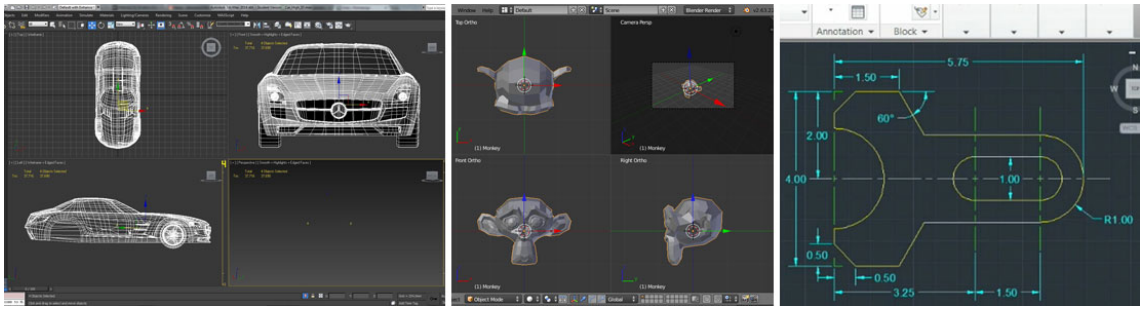


Figure 2.7: Orthogonal viewports in 3D Studio Max (left), Blender (middle) and AutoCAD (right).

2D manipulations along different axes, overcoming mapping issues. However, they require users to have greater spatial perception, rendering them suitable only for expert users. AutoCAD, which is more focused on architectural and engineering projects, also features these orthogonal viewports and allows for extremely precise manipulation of the elements within the VE.

2.2.1.3. Multi-DOF Controllers (non-tracked in 3D)

Several authors and companies have proposed advanced mouse-like devices allowing multi-DOF mapping on different hand actions on 3D rotation, translation and scaling.

SpaceMouse and other products by 3Dconnexion[1] are probably the most known examples and are also a commercial success, as they are used in CAD applications, visualization and are compatible with many related desktop application packages. These devices allow users to manipulate a pressure-sensitive handle to manipulate 3D models within an application. They allow to pan, zoom and rotate 3D objects simultaneously without external actions.

GlobeFish and GlobeMouse [45] are other experimental multi-DOF mapping devices. "The GlobeFish consists of a custom three degrees of freedom trackball which is elastically connected to a frame. The trackball is accessible from the top and bottom and can be moved slightly in all spatial directions by applying force. The GlobeMouse device works in a similar way. Here the trackball is placed on top of a movable base, which requires to change the grip on the device to switch between rotating the trackball and moving the base."

[1] 3Dconnexion: `www.3dconnexion.com`, accessed 3-January-2018.

CAT [53] is another experimental 6-DOF freestanding device. It consists of a round tabletop that can be rotated about its three axes and features a movable ring around it connected to dynamometers that able to check pressure applied in all three directions. Roly-Poly Mouse [100] attempts to combine the advantages of devices such as SpaceMouse for 3D pointing and manipulation tasks with the functions of a standard mouse when a 2D pointing task has to be performed.

2.2.2. 3D Manipulation on Interactive Surfaces

Beyond the traditional WIMP-based approaches, several multi-touch solutions to manipulate 3D objects have been proposed and evaluated over the past few years. In fact, touch-enabled displays have long been available, but their increased interest occurred following Jeff Han's work [54] and his acclaimed TED talk. With these interactive surfaces, new interaction possibilities emerged, allowing researchers to explore more natural user interfaces (NUIs) [132]. Efforts have been directed towards attempting to create more direct interactions with virtual content, closer to the ones with physical objects, which can successfully surpass mouse-based interactions [66]. Touch-enabled surfaces are now present in our everyday life through smartphones and tablets. Interactive tabletops are also becoming increasingly more popular. These types of surfaces have been used for a variety of purposes, including interacting with 3D virtual content.

2.2.2.1. Touching to Select

Unlike mouse-based interfaces, touch sensitive surfaces offer direct interactions with the 2D imagery. Instead of placing a cursor on top of the desired object and clicking, users can simply touch the object. This way selections on touch devices also have a screen-space reach, allowing users to interact with every object displayed. However, the user's finger is prone to occlude the virtual content (occlusion problem) and, since the finger's touch area is much larger than a pixel of the display, it can also lead to inaccurate selections (the fat finger problem) [131]. This can be overcome by using a cursor with offset or the middle point between two touches to perform selections on a multi-touch device [123], or showing a copy of the occluded screen area in a non-occluded location [126]. Some surfaces, either designed for collaboration or for increased visualization capabilities, are so large that its entirety is not within users' arm reach. To tackle this, the Vacuum technique [14] uses a single step refinement

strategy through a controllable widget that brings far away objects closer to the user, in the form of proxies, so that all objects displayed can be interact with.

2.2.2.2. Direct Touch Manipulations

To manipulate 3D objects using touch enabled surfaces, researchers initially proposed techniques for controlling several DOFs simultaneously, since it has been shown that rotation and translation have a parallel and interdependent structure in the human mind [130], Hancock et al. [55] developed techniques to control 6 DOFs using one, two and three touches. The authors started by extending the Rotate'N Translate (RNT) algorithm [71] to the third dimension. When touching an object, that object will follow the finger, rotating along all three axes and translating in two dimensions (Figure 2.8). Using two touches, the original two-dimensional RNT is used with the first touch, while the second touch rotates the object in the remaining axes. The distance between the two touches changes the depth of the object. The three-touch approach uses the first contact point for translations in a two-dimensional plane, the second to yaw and manipulate depth, and the third to pitch and roll. After evaluating this technique, the authors concluded that a higher number of touches provides both better performance and higher user satisfaction. These results suggest that a close mapping of input and output DOFs is desirable. The authors also defined a set of requirements for multi-touch interfaces, such as creating a visual and physical link with objects and providing suitable 3D visual feedback. Later, they improved the proposed techniques with Sticky Fingers and Opposable Thumb [56]. This solution is very similar to the three-touch technique, but in this solution, the third touch is used to rotate the object around the axis defined by the first two touches (Figure 2.9).

Considering the *de facto* standard for 2D manipulations, the Translate-Rotate-Scale (TRS) or two-point rotation and translation with scaling [57], Reisman et al. [107] proposed a screen-space formulation that uses several points of contact in a multi-

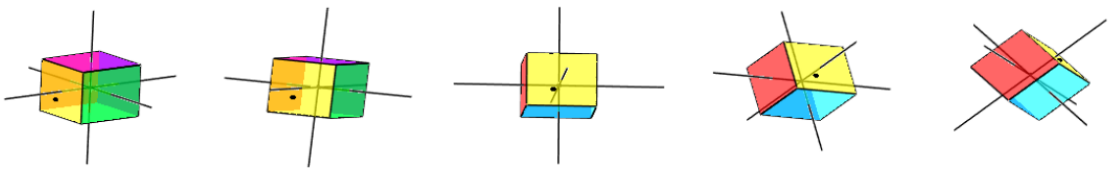


Figure 2.8: Shallow-depth single touch interaction: the object follows the touch (black dot), rotating along all three axes and translating in 2D (extracted from [55]).

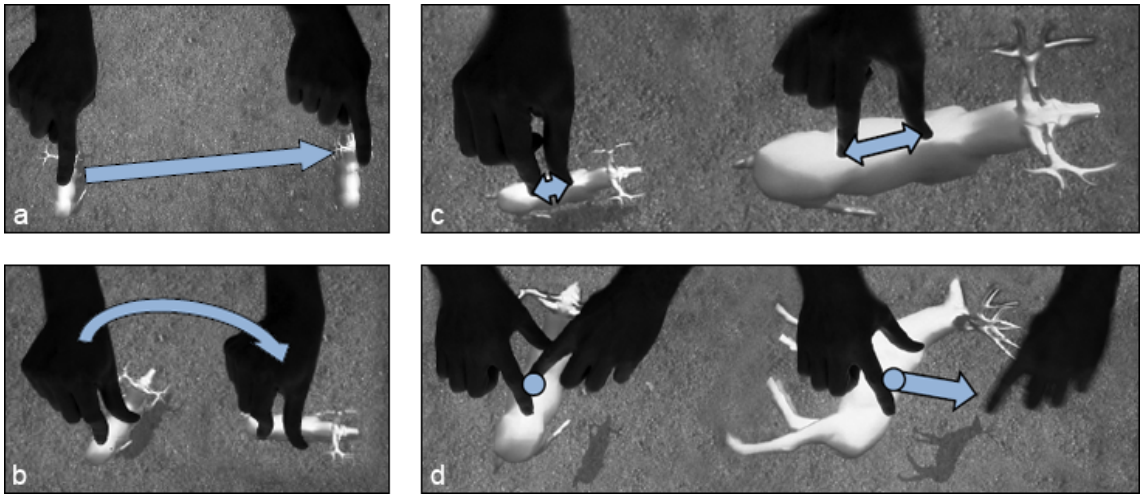


Figure 2.9: Sticky Fingers technique (a, b, c) and Opposable Thumb (d) (extracted from [56]).

touch device to manipulate 3D objects in 6 DOFs. Similar to previous works, rather than supporting scaling transformations, the distance between contact points is mapped to depth manipulation according to the view vector. The rationale is that the object appears larger when it is closer to the camera and smaller otherwise. Their solution keeps the contact points fixed throughout the interaction, using a constraint solver to move and rotate objects simultaneously. This solution is similar to Opposable Thumb, but if the movement of the third finger is not perpendicular to the defined axis, then that axis is no longer used and the object will rotate to follow the finger, as illustrated in Figure 2.10. The main issue of providing an integrated solution to manipulate different transformations simultaneously is that

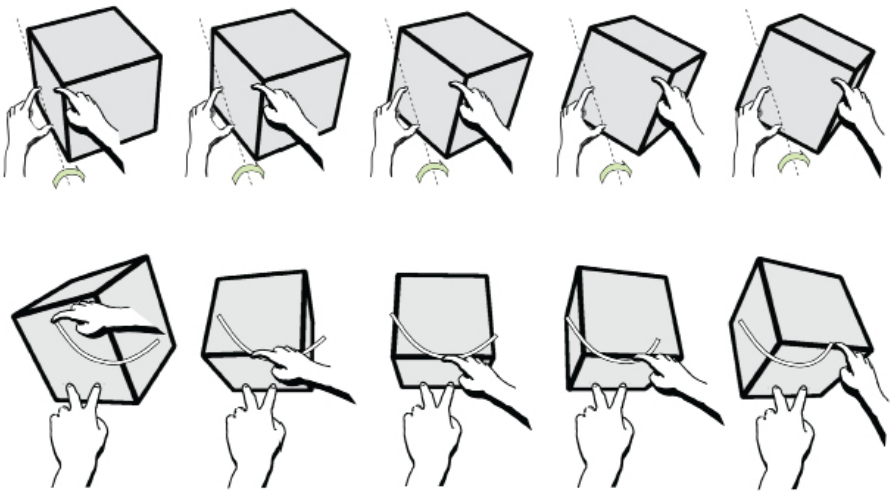


Figure 2.10: Screen-space formulation - two different rotations with three touches (extracted from [107]).

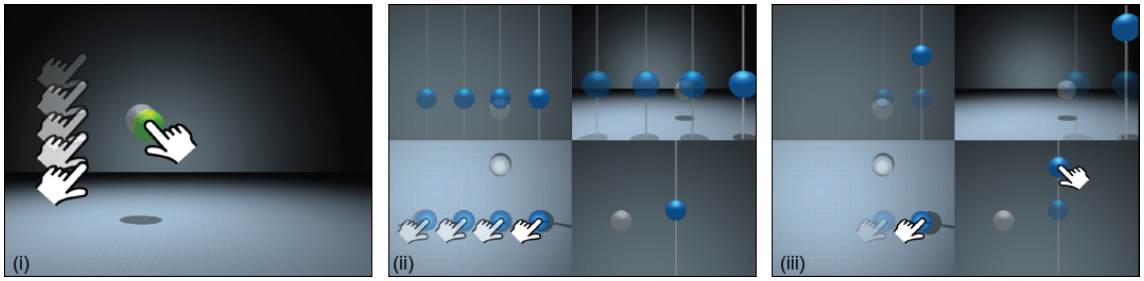


Figure 2.11: Z-Technique (i) and the orthogonal viewports approach (ii and iii). Gray lines indicate possible motions for the second touch (extracted from [81]).

unwanted operations arise frequently. To remedy this issue, the separation of DOF manipulation has been suggested [95] and followed in different research works.

Martinet et al. [81] proposed two techniques to translate 3D objects, as shown in Figure 2.11. The first extends the viewport concept found in many CAD applications (four viewports, each displaying a different view of the model). Touching and dragging the object within one of the viewports translates the object in a plane parallel to that view. Manipulating the object with a second touch in a different viewport modifies depth relative to the first touch. For the second method, denoted as the Z-technique, only one view of the scene is employed. In this technique, the first touch moves the object in the plane parallel to the view, while the backward-forward motion of a second touch is remapped to control the depth relative to the camera position. The authors' preliminary evaluation suggests that users prefer the Z-technique.

Improving upon the Z-Technique, Martinet et al. introduced DS3 [82], a 3D manipulation technique based on DOF separation. Similar to the Z-Technique, one touch moves the object in the screen plane, and an indirect touch manipulates object depth. Two direct touches in the object enable rotations, using a constraint solver similar to Screen-Space [107]. The authors compared DS3 with previous works [56, 107], and a user evaluation revealed that DOF separation led to better results. However, using a transformation plane parallel to the view plane can occasionally result in awkward transformations when the view plane is not orthogonal to one of the scene axes [87].

Rather than using the number of users' touches to determine the type of transformation to apply, Liu et al. [75] use the movement characteristics of two touches (Figure 2.12). Two moving touches control 4 DOFs (3 translation and 1 rotation) in a manner similar to Sticky Fingers. One fixed touch and another moving touch control the remaining 2 DOFs. Although outperforming the screen-space and DS3 approaches and being comparable to Sticky Fingers while requiring less contact

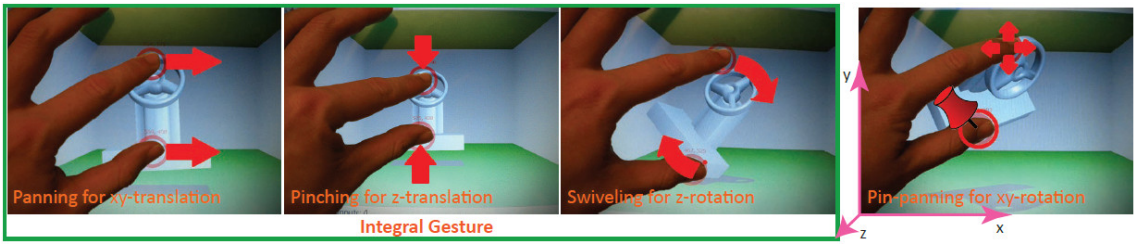


Figure 2.12: Two-finger gestures for 6-DOF manipulation: xy-translation, z-translation, and z-rotation controlled by an integral RST-style gesture and xy-rotation controlled by pin-panning gesture (extracted from [75]).

points, the authors state that their technique might not be very suitable for fine-tuning control of object transformations.

2.2.2.3. Indirect Interactions through Input Remapping

As we previously presented for mouse-based manipulations, a common approach for input remapping is the use of virtual widgets. Schmidt et al. [109] introduced a 3D manipulation approach for sketch-based interfaces, combining 3D widgets, context-sensitive suggestions and gestural commands. Users indicate an object to transform by explicitly selecting it with a tap, and by drawing a stroke, the system responds by automatically creating translation and rotation widgets based on the candidate axis nearest to the stroke. Candidate axes include world and object axes. Initial widgets can be modified using context-sensitive gestures or by drawing a different axis.

To better understand user gestures for 3D manipulation tasks on multi-touch devices, Cohé et al. [31] conducted a user study and concluded that physically plausible interactions are favoured, and there are different strategies to develop an application focusing on broad usage or ease of use. Based on observations of users interacting with widgets for 3D manipulations, Cohé et al. [30] designed a 3D transformation widget called tBox. This widget allows the direct and independent control of 9 DOFs (translation, rotation and scaling along each axis). tBox consists of a wire-frame cube, which is visible in Figure 2.13. Users can drag an edge of the cube to move the object in an axis containing the edge, and rotations are achieved by dragging one of the cube's faces.

To create VEs for computer-animated films, Kin et al. [67] designed and developed Eden, a fully functional multi-touch set construction application (Figure 2.14). Virtual objects can be translated in a horizontal plane using the usual direct drag

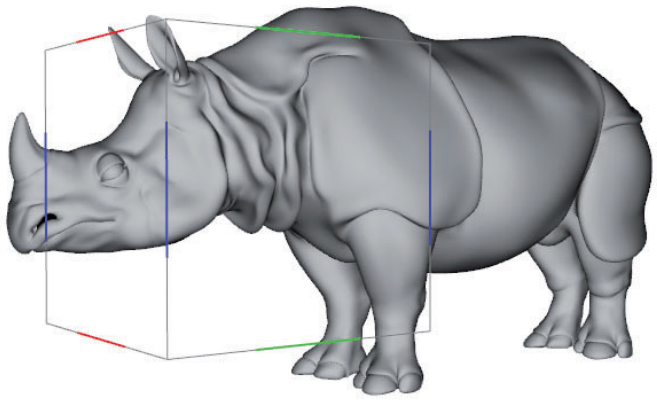


Figure 2.13: The tBox widget (extracted from [30]).

approach and up and down with a second finger, similar to the Z-technique [81]. Rotations are performed similar to the Arcball [111] widget. It also supports both uniform and one-dimensional scaling transformations.

LTouchIt [88], although using direct manipulation for translations, also relies on widgets for rotations. Following the DOF separation, it has a set of interaction techniques that provide direct control of the object's position in no more than two simultaneous dimensions and rotations around one axis at a time using rotation handles. The translation plane is perpendicular to one of the scene axes and is defined by the camera orientation. Using the rotation handles, the user can select a handle to define a rotation axis and, with another touch, specify the rotation angle, as exemplified in Figure 2.14.

Au et al. [5] use the high input bandwidth of multi-touch surfaces and delegate the manipulation power of standard transformation widgets to multi-touch gestures. This enables seamless control of constraint and transformation manipulation using a single multi-touch action (Figure 2.15). Users can select a candidate axis with two touch points, and transforming the object is performed by holding and moving two

Figure 2.14: Placing (left) and rotating (middle) objects in Eden (extracted from [81]), and LTouchIt's rotation handles (right) (extracted from [88]).

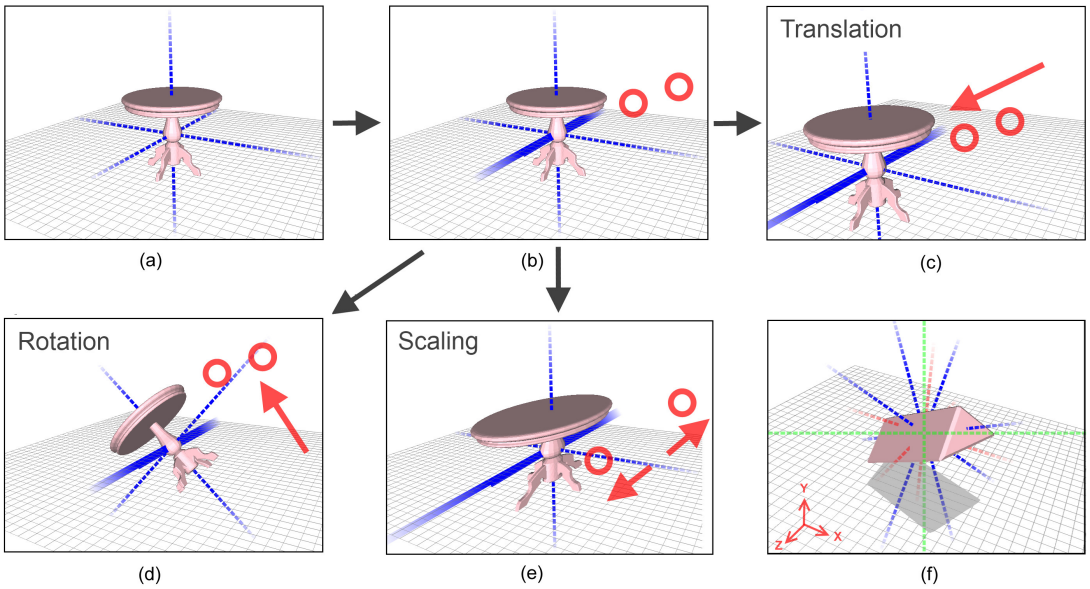


Figure 2.15: Multi-touch gestures for axis-based manipulations: (a) virtual object; (b) axis selection with two touches; (c)-(e) axis-constrained translation, rotation and scaling; (f) total set of candidate axes (extracted from [5]).

fingers.This approach also supports plane constraints by using a candidate axis as the plane normal and transformations relative to a pivot point located on another object.

Regarding direct versus indirect interactions, Knoedel et al. [69] investigated the impact of the directness in TRS manipulation techniques. Their experiments indicated that a direct approach is better for completion time but that indirect interaction can improve both efficiency and precision.

Bollensdroff et al. [16] redesigned older techniques for three-dimensional interactions [60] using multi-touch input. They developed a cube-shaped widget, the Gimbal Box, which uses a touch in one of its faces to translate in the plane defined by that face (Figure 2.16.a). To rotate the object, the widget has two variations. One

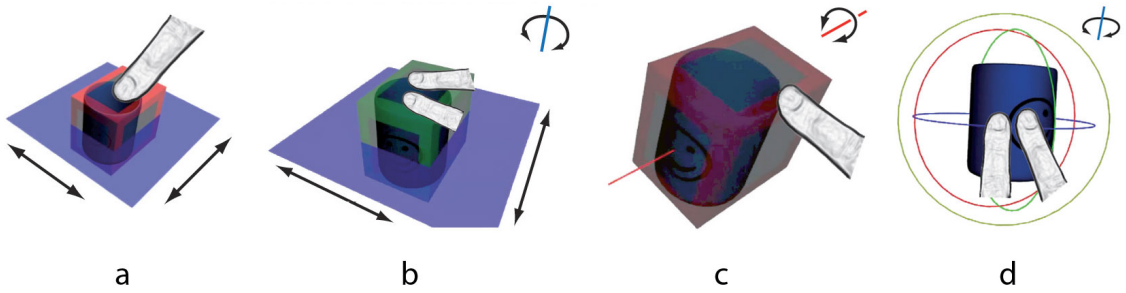


Figure 2.16: GimbalBox - translation (a) and different approaches for rotation (b, c, d) (extracted from [16]).

uses the TRS applied to a cube's face (Figure 2.16.b); alternatively, touching an edge of the box induces a rotation around an axis parallel to the edge (Figure 2.16.c). The other variation is based on Arcball [111] (Figure 2.16.d). Through a controlled study, their techniques were compared to other approaches that are well known in the literature [56, 107]. They concluded that adapted widgets are superior to other approaches for multi-touch interactions, supporting DOF separation through the reduction of simultaneous control to 4 DOFs in a defined visible 2D subspace. Moreover, the authors suggest that "multi-touch is not the final answer" since "the projection of an object as input space for interaction can never reproduce precise motions of the object in 3D space".

TouchSketch [135], an interface for editing the shape of 3D objects, divides object manipulation into three categories: axis constrained, plane constrained and uniform manipulation. For this purpose, it resorts to a constraint menu, which allows users to select a constraint in the menu with the non-dominant hand and use the dominant hand to apply transformations respecting the selected constraint (Figure 2.17). Evaluation results suggest that this technique can outperform a single-touch approach based on widgets in terms of efficiency.

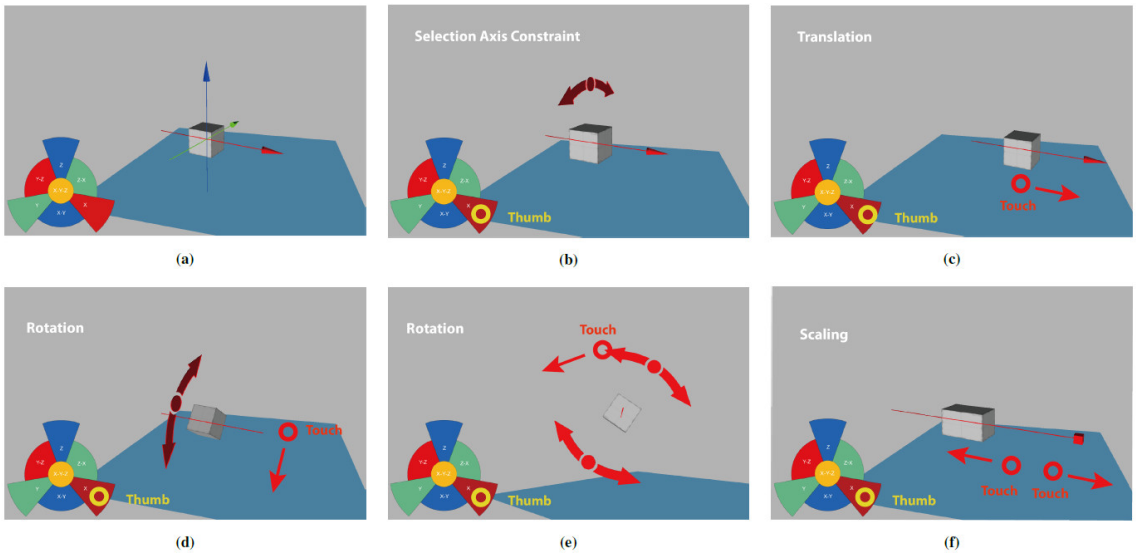


Figure 2.17: TouchSketch manipulations: (a) initial state; (b) X-axis constraint specified; (c)-(f) translation, rotation and scaling according to the constraint (extracted from [135]).

2.2.2.4. Touching Stereoscopic Tabletops

To improve both three-dimensional visualization and spatial perception, several researchers have explored interactions using stereoscopic environments. In such environments, since virtual objects can appear outside the surface, either in front of or behind the surface, previous touch techniques are not suitable. Directly touching the surface where the object is projected can disrupt the illusion and be unnatural, thus the need for different manipulation techniques. Considering the placement of virtual objects inside the tabletop in a fish-tank approach, touch solutions suffer from parallax issues [93]. Above the table solutions have already been explored. Using the Responsive Workbench, one of the first stereoscopic tabletop VR devices, Cutler et al. [33] constructed a system that allows users to manipulate virtual 3D models with both hands. The authors explored a variety of two-handed 3D tools and interactive techniques for model manipulation, constrained transformations and transitions between one- and two-handed interactions. However, they resorted to toolboxes to allow the user to transition between different operations.

Benko et al. [10] proposed a balloon metaphor to control a cursor (Figure 2.18), which is then used to manipulate three-dimensional virtual objects on a stereoscopic tabletop. Moving two fingers closer, the user allows the object to move up, and likewise, if the user moves the fingers away, the object will translate downwards. Later, Daiber et al. [34] created a variation of this technique by adding a corkscrew metaphor, which can be used with either both hands or a single hand. With this approach, the user can use a circular motion in a widget rather than the distance between fingers to manipulate an object's height. The authors compared their technique with the previous techniques in both positive and negative parallax scenarios. Although none of the techniques was clearly identified as being better, the negative parallax space was shown to be more difficult to interact with.

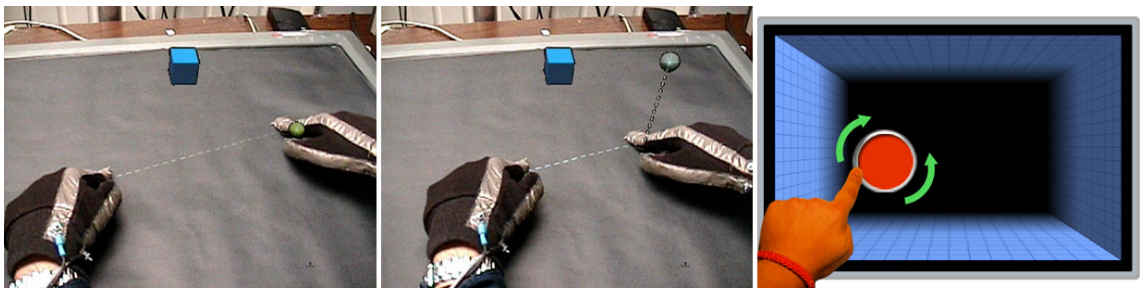


Figure 2.18: The balloon metaphor (left and middle): moving two fingers closer translates the cursor upwards (extracted from [10]). Corkscrew variation (right): circular motions replace the distance between touches (extracted from [34]).

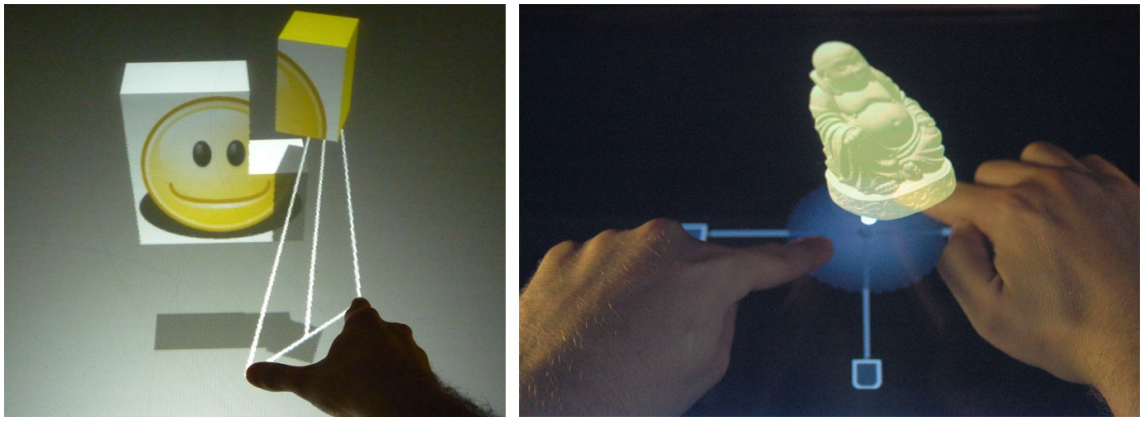


Figure 2.19: Left: triangle cursor (extracted from [121]). Right: Toucheo interaction (extracted from [52]).

Strothoff et al. [121] proposed another approach to select and manipulate a cursor in stereoscopic tabletops. Using two fingers to define the base of a triangle, the height of the cursor, placed in the third vertex, is defined by the distance of the two touches, as exemplified in Figure 2.19. Using this triangle cursor, users can manipulate selected objects in 4 DOFs: translation in three dimensions and rotations around a vertical axis.

To manipulate virtual objects in the full 9 DOFs, Toucheo [52] presented a setup with co-located 3D stereoscopic visualization, allowing people to use widgets on a multi-touch surface while avoiding occlusions caused by the hands. The authors combined a two-dimensional TRS interaction on the surface with the balloon metaphor [10] and other widgets that provide both the remaining rotations and independent scaling along three axes.

Previous works [10, 34, 121, 52] prevent the vergence-accommodation conflict, which can lead to the loss of the stereoscopic effect or cause discomfort, by touching below the virtual object in the stereoscopic display. Simeone [112] followed a different approach based on indirect touch interaction through an additional multi-touch surface. The author proposed two novel indirect manipulation techniques, Indirect4 and Indirect6, one to control 4 DOFs and the other for 6 DOFs, respectively. The first uses a touch from the dominant hand to control horizontal translations and a touch from the non-dominant hand to modify the object's vertical position (with vertical motions) and rotation around a vertical axis (with horizontal motions). The second technique manipulates the object's position similarly, but it uses two touches from the non-dominant hand to perform rotations. If the two fingers move horizontally or vertically, yaw or pitch is enabled, respectively. If they move in opposite directions, roll is enabled. These techniques were compared to DS3 [82] and

Triangle Cursor [121]. The results showed that indirect touch interaction techniques provide a more comfortable viewing experience while presenting no drawbacks when switching to indirect touch.

Giesler at al. [47] proposed the Void Shadows technique for fish tank stereoscopic tabletops. This technique offers control over 4 DOFs (3 for translation and 1 for rotation) for each object present in the VE. Each object projects a fake shadow on the zero parallax plane, and the user is able to touch it directly. Direct translation from the finger position is applied to the object on the XY plane controlling 2 translation DOFs, while translation on the Z-axis is performed with a pinch gesture. The only rotation available is about the Z-axis and is performed by rotating two or more fingers in contact with the shadow around its centre. This technique allows all these interactions to be performed simultaneously if the user wishes to do so.

2.2.3. Mid-Air Interactions

Mid-air interaction, e.g. based on a spatial input realized in a physical 3D context, provides the potential to manipulate objects in 3D with more natural input mappings. This type of interaction is enabled by tracked handheld devices (or wearable devices) or by tracking users' hands with external sensors (e.g. cameras, depth cameras).

2.2.3.1. Enabling Technologies: Handheld Devices and Hand Trackers

Using inertial sensors, computer vision or magnetic tracking, the orientation and position of a handheld device can be derived and used for controlling virtual objects. Tracked handheld devices are the current solution proposed by the gaming VR industry with well-known commercial products such as Nintendo Wii, Playstation Move [2], HTC Vive[3], and Oculus Touch[4]. These devices can provide at least 6-DOF tracking capabilities per controller, with extra DOFs depending on the number of buttons and control sticks that each controller possesses. In addition to being more suitable for video games and similar interactive applications with their button layout

[2]PlayStation Move: `www.playstation.com/en-us/explore/accessories/vr-accessories/playstation-move`, accessed 3-January-2018.
[3]HTC Vive: `www.vive.com`, accessed 3-January-2018.
[4]Oculus Rift: `www.oculus.com/rift`, accessed 3-January-2018.

resembling those of standard gaming controllers or television remote controllers, they are also easier to track when compared to human body parts such as hands, limbs or heads.

Handheld manipulation devices can also be everyday life items, such as phones. In [63], it is shown that the 3-DOF orientation sensor of a phone can be effectively applied for controlling the orientation of a 3D virtual object.

Specific handheld devices have been designed for specific interaction and manipulation tasks. The Cube Mouse [46] is a 6-DOF tracked object with three rods that can be pulled and twisted, mapping other translational and rotational controls. This mouse was designed for specific visualization tasks supporting a bimanual control for moving and slicing objects.

The advantage in using these types of devices is that they allow users to both use a virtual hand paradigm by mapping the 6 DOFs provided by tracking in space position and rotation directly to a virtual hand in a VE and to map grabbing actions, scaling, and eventually rotational and translational DOFs to buttons and other controller devices such as joysticks and touchpads. This avoids the use of gesture or voice recognition algorithms required by deviceless setups to enable multiple actions.

Furthermore, modern technologies allow for the addition of some types of haptic feedback on the devices that can be used to add realism to the interaction. For instance, the Oculus Touch and HTC Vive controllers can provide haptic feedback through controlled and tunable vibrations, allowing different feedback channels and potentially freeing space in the virtual scene, avoiding unnecessary cluttering.

However, despite the high potential and the choice of these types of devices by low-cost HMD-based solution developers, the use of smart controllers does not solve, per se, the manipulation issues related to smart mapping of user actions into object transforms, control of rotation and scaling, out-of-reach objects, and accuracy.

More freedom than holding a device could be achieved using wearable devices (e.g. gloves [38]). The Color Glove [129] enables precise finger and hand pose tracking. The system uses a simple RGB camera to capture the coloured areas of the gloves, being able to reconstruct the user's entire hand, thereby obtaining full 6-DOF tracking in real time. Manipulation based on wearable devices has been tested [28]; however, such tests showed that their use does not solve the usability issues of freehand manipulation, requiring the development of smart metaphors and feedback solutions. Furthermore, low-cost hardware solutions are not yet available.

A different approach in mid-air interaction is based on tracking hands without the need for handheld objects, exploiting depth sensors such as Microsoft Kinect or visible or IR stereo cameras (e.g. in the LeapMotion sensor). Wang et al. [128] introduced a new way to track hands and fingers using affordable depth cameras. Their approach, in addition to pose detection, tracks each hand in 6 DOFs in a non-invasive manner. These tracking solutions allow hand reconstruction, which can be used to closely mimic physical interactions. The possibility of tracking fingers opens several possibilities [23], but the tracking performances are not always satisfactory.

2.2.3.2. Grabbing to Select

Most approaches for object manipulation in mid-air resort to a natural grab metaphor. Users move their hands or representative cursor close to the desired object, typically intersecting it, and perform a grab action, which can be to close the hand or to press a button, for example. The object is then selected and further actions are applied to it. Naturally, this approach has an arm-length reach. Other approaches that tackle this limitation are presented in Section 2.2.3.5. For cluttered environments, since the bounding volume of the hand or cursor can intersect several objects at the same time, the Intent-Driven Selection [101] proposes the use of a scalable sphere as bounding volume, allowing for a continuous progressive refinement until the desired object is selected.

2.2.3.3. Mappings and Metaphors

Allowing for more natural input mappings than those offered by mouse and touch interfaces, interactions such as grab, move and rotate objects can be performed in in mid-air, similarly to how they are performed with physical objects [108]. For instance, the HoloDesk [59] allows direct interaction with 3D graphics using physics simulation, resorting to a setup similar to Toucheo [52] and a depth camera for hand tracking (Figure 2.21).

However, it is more common to pre-program interactions, explicitly mapping input DOFs onto transformation DOFs. With the Simple Virtual Hand manipulation [73], users can directly grab objects, and they will follow the grabbing hand until released. All hand movements are applied to the object: dragging moves the object, and hand rotations change object orientation. This approach is natural, but it can

be challenging or not effective for some applications due to the limited range of translation and rotation and lack of precision [23]. Additionally, not always a full 6 DOF tracking of users' hands is available. For instance, several tracking solutions are only capable of 3 DOF positional tracking, without supporting orientation. For these reasons, a vast amount of research has been dedicated to proposing effective solutions for motion mapping and manipulation metaphors.

Hilliges et al. [58] presented a technique to seamlessly switch between interactions on the tabletop and above it. The main goal of the authors was to create a solution that resembles physical manipulations, enabling depth-based interactions. Using computer vision, the user's hand is tracked in 4 DOFs (3 for translation and 1 for rotation), and the grab gesture can be detected. Shadows of the user's hands are projected into the scene, which are used to interact with virtual objects in three dimensions. After an object is grabbed by the user's shadow, the modifications in the corresponding hand are applied to the object, as exemplified in Figure 2.20.

Marquardt et al. [80] also combined the multi-touch surface and the space above it in a continuous interaction space. Taking advantage of this space, they leveraged the user's hands movements to allow full 6-DOF interaction with digital content. Following this continuous space, Mockup Builder [2, 3] offers a semi-immersive modelling environment in which users can freely manipulate three-dimensional virtual objects. The authors used GameTrak devices to follow the positions of users' fingers in 3 DOFs, which acted as cursors, and adapted TRS to three dimensions to manipulate objects in mid-air with 7 DOFs (we will refer to this technique as Air-

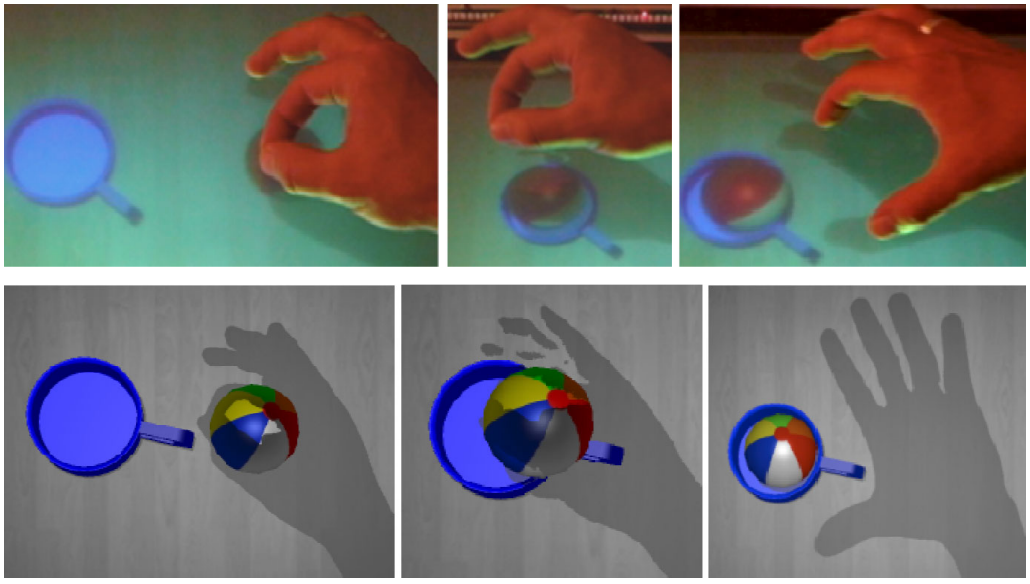


Figure 2.20: Virtual shadows used to manipulate an object (extracted from [58]).

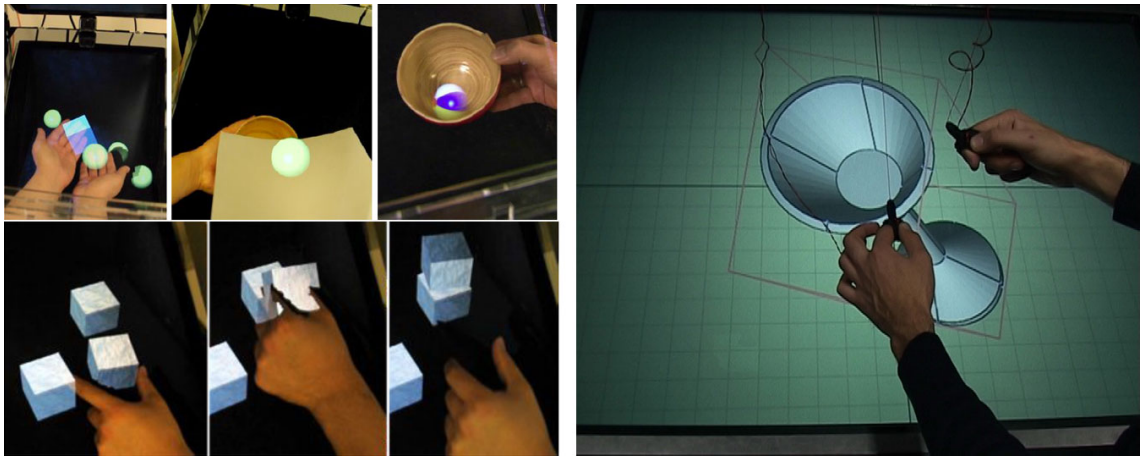


Figure 2.21: Left: users interacting with HoloDesk (extracted from [59]). Right: User scaling an object in Mockup Builder (extracted from [3]).

TRS). With one hand users can directly grab and move an object, while a second hand, after performing a grab gesture outside the object, allows rotations around the first. Additionally, the distance between both hands is used for uniform scaling operations.

Kim and Park [65] proposed a Virtual Handle with a Grabbing Metaphor (VHGM). When the user selects an object, the system generates a bounding sphere around the object. From the sphere's centre, a ray with its direction opposite to that of the virtual handle is projected to find the intersecting point on the sphere (Figure 2.22). This point serves as the reference frame for the following transformations (transla-

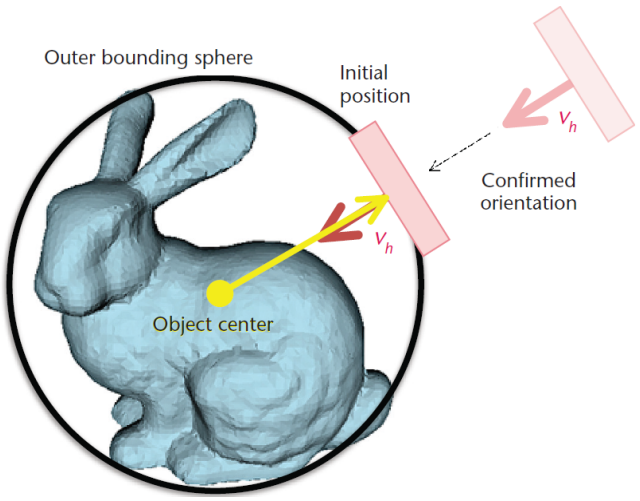


Figure 2.22: Virtual Handle with a Grabbing Metaphor: finding handle's initial position. When the user selects an object, a bounding sphere is generated around the object (extracted from [65]).

tion and rotation). User evaluation results suggest that VHGM can lead to better rotation efficiency than a standard 3D cursor.

Mapes and Moshell [79] introduced Spindle, a bi-manual technique to manipulate virtual objects. The point between user's hands is used to select the object and acts as the transformation center. Moving both hands at the same time in the same direction makes the object to translate, and moving them around the center rotates the object accordingly. Changing the distance between both hands scales the object. While Mapes and Moshell [79] used specific gloves as input devices, Bettio et al. [13] latter implemented this technique using computer vision, thus allowing a hands-free interaction.

Song et al. [116] proposed a Handlebar metaphor (Figure 2.23), an approach similar to Spindle, using a single depth camera to track the position of users' hands in space. Since users' hands are only tracked in 3 DOF, rotations around the axis of the line defined by both hands can be achieved with an isolated swivel gesture. This technique also allows users to manipulate single objects or pack multiple objects along the handlebar. More recently, Cho et al. [29] proposed Spindle+Wheel, also based on Spindle and similar to the Handlebar, developed for semi-immersive environments and resorting to spherical handheld devices for hand tracking. This approach uses an offset between users' hands and virtual cursors, and it is also two handed. Moving both hands in the same directions translates the object between them, and moving both hands in different directions rotates the object (roll and yaw). Changing the distance between hands performs scaling operations, and rotating one of the hands rotates around the main axis of the handheld device (pitch). The main difference from the Handlebar technique is that Spindle+Wheel offers simultaneous 7-DOF transformations.

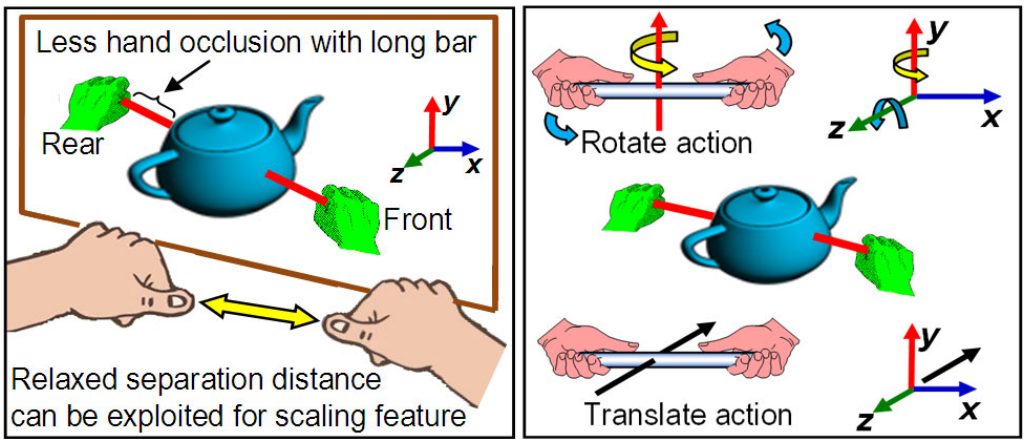


Figure 2.23: The Handlebar metaphor used to translate, rotate and scale a virtual object (extracted from [116]).

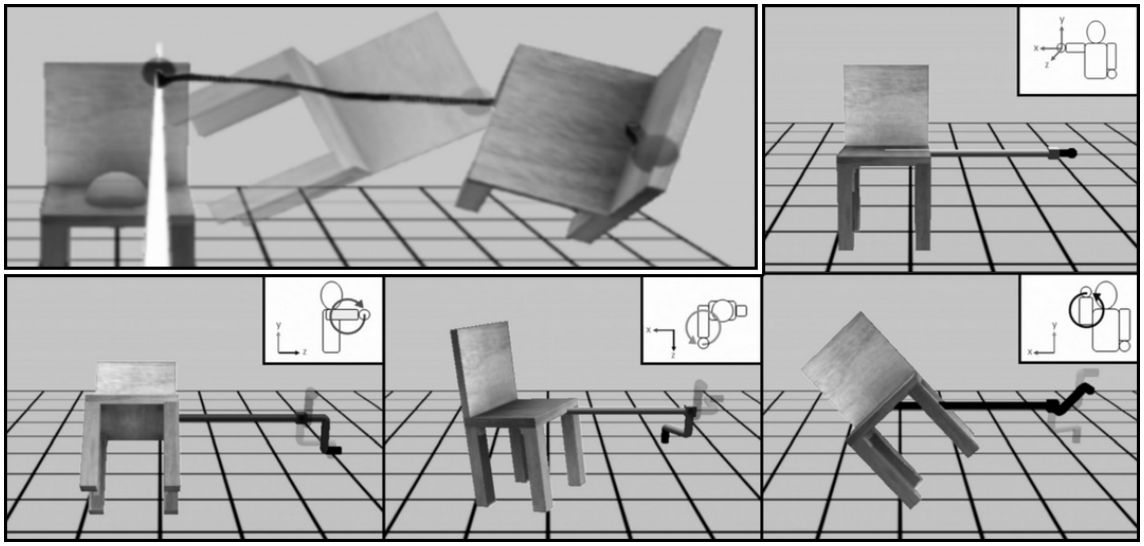


Figure 2.24: Top left: Grasping Object. Sequence of transformations following the black path. Top right and bottom: Crank Handle. From left to right: translation mode, rotation mode X-axis, rotation mode Y-axis, and rotation mode Z-axis. (extracted from [18]).

Bossavit et al. [18] proposed two manipulation techniques: the Crank Handle (CH) and the Grasping Object (GO). The CH is a one-hand technique that separates translation from rotation and decomposes rotations in primary axes. These axes are selected through a crank handle metaphor (Figure 2.24). The GO technique is another one-hand manipulation technique. In contrast to the CH, it combines translation and rotation and does not decompose rotation in the primary axes. The authors based this technique on the RNT algorithm and on its 3D extension for 2-DOF inputs and extended it further to support 3-DOF positional input.

2.2.3.4. Mobile-device-based Mappings and Metaphors

When a tracked object is available, mappings can be enhanced by the use of objects' orientations and specific input channels. Berge et al. [12] proposed a classification for this specific manipulation technique that uses a common smartphone as a smart object to interact with a VE. The classification is based on three categories: around the smartphone (ASP), with the smartphone (WSP) and on the smartphone (OSP). Whereas the OSP category simply includes all the mere implementations of touch-based techniques on the phone screen with the sole difference of the effect of the interaction occurring remotely, the ASP and WSP categories offer a tool to classify techniques with an emphasis on the role played by the smartphone in the interaction. In WSP techniques, the smartphone is a traditional smart object used directly as

a reference by the system to track the user's hand movement. Meanwhile, ASP techniques use the smartphone position as a reference frame for the dominant hand and its screen to provide visual feedback for the user.

Issartel et al. [62] analysed the manipulation of virtual objects through the use of a mobile device and the way the movement is mapped between the two. Three categories are presented: absolute position control, relative position control and rate control. The work offers insights on the benefits and disadvantages of the different solutions along with a more in-depth study on the implications caused by factors such as spatial feedback compliance and allocentric/egocentric design choices.

Speicher et al. [118] implemented a combined technique using both a Microsoft Kinect and a mobile phone to manipulate virtual objects. The Microsoft Kinect was used to track 3 DOFs for the hand position, while the mobile phone held in the user's dominant hand was used to track 3 DOFs for the rotation. The interaction technique was validated with a docking task and with measurements of the task completion time, translation task precision and rotation task precision. These last two measures were further subdivided by taking into account performances on the three different axes.

Mine et al. [89, 90] converted the desktop application SketchUp into a virtual reality application: VR SketchUp (Figure 2.25). Their objective was to develop interaction techniques that can run across a spectrum of displays, ranging from the desktop and head-mounted displays to large CAVE environments, minimizing energy while maximizing comfort. For this purpose, they constructed a hybrid controller that collocates a touch display and physical buttons through a 6-DOF tracked smartphone attached to a handheld controller. 3D spatial input was used to achieve a coarse starting step. Meanwhile, 2D touch was used for precision input, such as controlling widgets, defining constraints and specific values for transformations, and

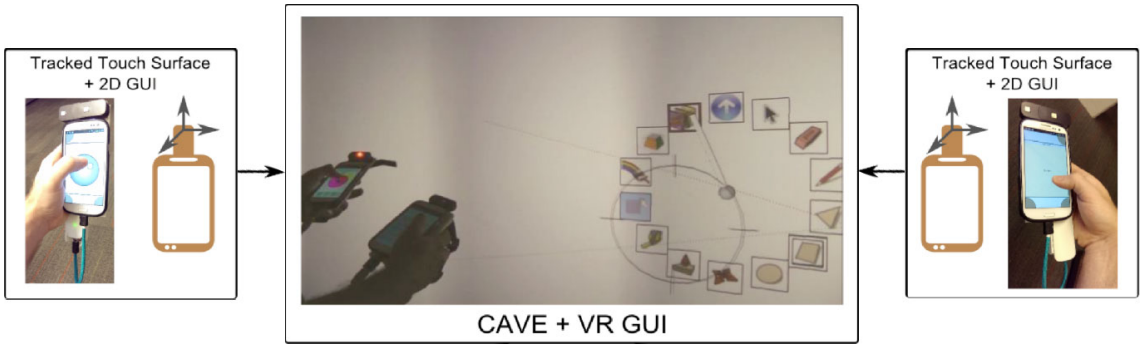


Figure 2.25: VR SketchUp interface. Middle: floating VR GUI, left: non-dominant hand controller, right: dominant hand controller (extracted from [89]).

providing numeric or textual input. To manipulate objects, the authors presented three alternatives: direct 6-DOF manipulation, where scaling of the object can be achieved using bi-manual interaction and DOF constraints, rotational axes, and special behaviour such as position-only manipulation are specified using the touch-screen interface; image plane interactions, where movement of the user's hand within their field of view is mapped to screen space interactions; and trackpad interaction, where the user manipulates objects via a touchpad widget on the touch screen to emulate mouse interactions within the user's screen space. Although the authors focused on several types of displays, resorting to imagery on the smartphone screen may not work well in conjunction with HMDs. However, some interactions on the touch surface were designed to not require the user to have to look down at them, such as menu navigation, which is represented by floating graphical elements in the VE.

2.2.3.5. Interacting Outside Arm's Reach

One of the first challenges addressed concerning the manipulation of virtual objects in mid-air was how to extend users' capabilities by allowing interactions with objects that are out of reach of users' arms. One of the first approaches to perform this type of interactions was ray-casting. It allows users to easily select and manipulate objects simply by pointing at them. This technique was used in the seminal work Put-that-there [17] to select objects, using a voice command to effectuate the selection. This trigger event can be changed, for instance, for the press of a button. The pointing gesture can be converted into a vector or ray in the virtual environment, typically attached to the user's hand tracked in 6 DOF. The first object it hits can be selected. Then, the object can be attached to the end of the pointing ray in order to be manipulated, moving and rotating accordingly [73].

The Go-Go immersive interaction technique [103] uses an arm-extension metaphor, interactively growing the user's arm, and a nonlinear mapping for reaching and manipulating distant objects (Figure 2.26). When the user's hand moves above a certain distance, the arm grows according to a predefined coefficient. Below that distance, a 1:1 mapping is used. This technique allows for seamless direct manipulation of both nearby objects and those at a distance. The Stretch Go-Go [20] technique improves on the previously mentioned Go-Go, by being able to extend the virtual arm until infinite. This achieved by continuously extending the virtual arm instead of using a scaling coefficient.

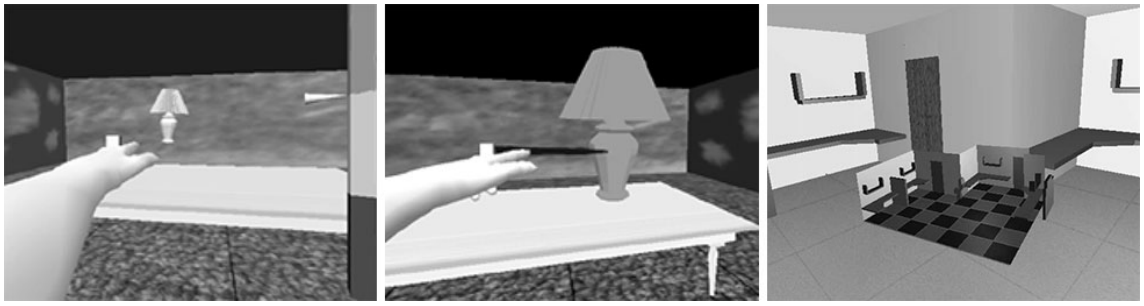


Figure 2.26: The Go-Go (left), ray-casting (middle) and Worlds in Miniature techniques (right) (extracted from [20] and [120]).

However, when comparing the Go-Go technique with other approaches, such as Stretch Go-Go and ray-casting, there is no clear winner [20]. User evaluation results showed significant drawbacks in all techniques. From this evaluation, HOMER (Hand-centered Object Manipulation Extending Ray-casting) was proposed. It uses ray-casting to select the object, and after selecting the object, it moves the virtual hand to the object. The current distance between the user's body and hand is mapped to the distance to the virtual object. Therefore, manipulation is performed similarly to the Go-Go technique, but the scaling coefficient is calculated for each selected object.

Lock Ray and Depth Ray [48] also improve ray-casting, but address only object selection. Since the ray can intersect several objects at different depths, they use forward and backward hand movements to continuously disambiguate between the intersected objects. In Lock Ray these operations are performed in sequence, while in Depth Ray they are performed simultaneously.

Ray-casting also has the downside that a small movement of users' hands can move the ray away from the target object. This is aggravated by the fact that objects far away appear small. To tackle this, Flashlight [74] uses a cone as selection volume instead of a ray to select a group of objects. It then uses a single step automatic refinement based on the objects' proximity to the origin of the cone, selecting the closest one. The Aperture and Orientation technique [41] improves the Flashlight by resorting to a scalable cone originated at the eye of the user as a selection volume, and uses the proximity to cone origin and selectable objects' rotation as disambiguation parameters. The shadow cone-casting [119] also uses a cone for selecting multiple objects but, for the disambiguation, the origin of the cone must be moved while maintaining the desired objects inside the cone. The final disambiguation is based on proximity to the origin of the cone. The EiHCam [84] uses a scalable truncated pyramid attached to a tracked tablet as selection volume. The disambiguation is done by a single discrete step, using the rendered image of the virtual camera with a

screen-space metaphor on the tablet screen. The Disambiguation Canvas [37] uses a mobile device to select objects in highly cluttered scenes. This process is composed by sphere-casting using the device sensors and the second using a representation of the touch screen in the IVE to select multiple objects also in a single step.

Still targeting disambiguation in mid-air selections, but designed for traditional displays, the Zoom [26] approach employs a zoom operation on a specific region of the environment to ease selection of small or distant objects. This metaphor is expanded on the Continuous Zoom [9] technique, where the scene is zoomed in until the target is large enough for selection. The Discrete Zoom [9] works similarly, but uses instead several discrete operations. These operations consist in dividing the screen in four quadrants and expanding the selected quadrant until the object of interest is large enough for selection. The SQUAD technique [70], on the other hand, uses several discrete steps to iteratively select an object within a group of interest. This technique uses a first phase where users cast a sphere to specify a volume containing the target object. Users may then disambiguate the selection in one or more phases using a menu that distributes the objects in four groups based on their characteristics. The Expand technique [26] uses a similar approach, but resorts to virtual grids which are iteratively rearranged until the desired object is selected. Regarding efficiency, both Zoom and Expand techniques performed better than the standard ray-casting and SQUAD techniques, and the Expand technique also provided better selection times than the Zoom [26]. Continuous Zoom technique achieved better selection times when compared to Discrete Zoom, SQUAD and ray-casting [9].

Other approaches for manipulating out-of-reach objects in large IVEs are the Worlds in Miniature [120] and Voodoo Dolls [102]. With Worlds in Miniature [120], all objects in the 3D environment can be accessed. Users control a miniature copy of the environment and can select objects from this representation within arms reach. Even though the whole environment is available, there is a limit to its size, so we consider this techniques' reach as scaled. Focusing on manipulating objects regardless of their scale, Pierce et al. [102] proposed the Voodoo Dolls technique, which dynamically creates scaled handheld copies of the objects (dolls) that can be manipulated rather than the objects themselves. Dolls can be created even for objects outside arm's reach, using different approaches (Figure 2.27). These dolls are used in pairs, one in each hand, and their effect depends on whether they are held in the right or left hand: the right hand's object is positioned in relation to the left hand's object. With this technique, users can work at multiple scales without explicitly resizing objects or the world.

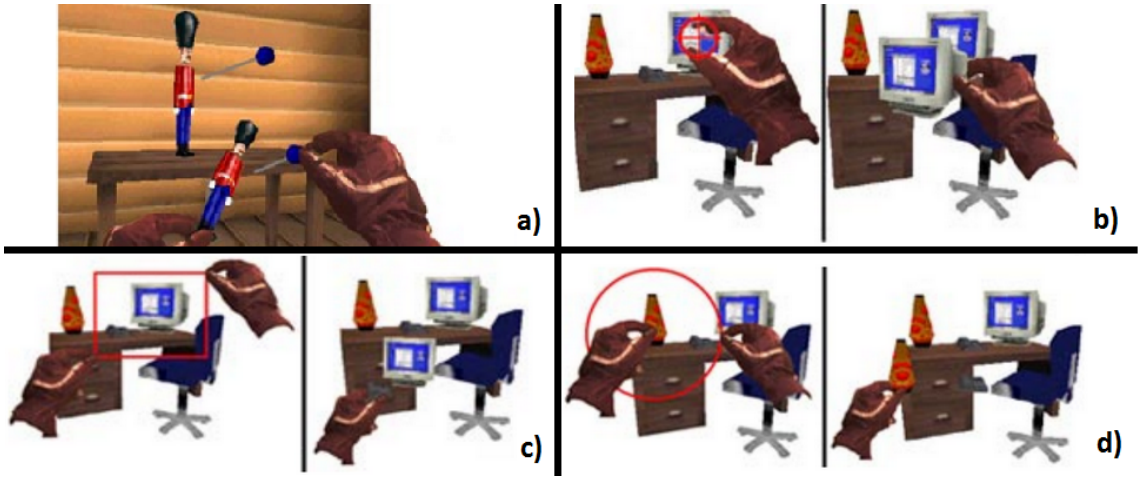


Figure 2.27: Voodoo Dolls technique: (a) manipulating a pin and toy soldier with dolls; (b) creating a doll; (c) framing the desired context; (d) specifying the radius for the context (extracted from [102]).

2.2.3.6. Solving Precision Issues

To overcome the lack of precision with object positioning techniques in immersive VEs, Kiyokawa et al. [68] proposed manipulation aids consisting of discrete placement constraints (snapping) and collision avoidance mechanisms. Without imposing placement restrictions, Frees et al. [42] introduced the PRISM (precise and rapid interaction through scaled manipulation) technique. In contrast to techniques such as Go-Go, which scale up hand movement to allow long-distance manipulation, PRISM scales the hand movement down to increase precision. Switching between precise and direct modes occurs according to the current velocity of the user's hand, as exemplified in Figure 2.28. When moving an object from one general location to another, the user is not necessarily interested in being precise and moves relatively

a. User moves hand slowly to the right and down. Some movement is scaled down in the horizontal direction, nearly all is scaled down in the vertical.

b. User moves in the same direction as in (a), this time quickly. Since interaction is in direct mode, object follows hand to its exact location.

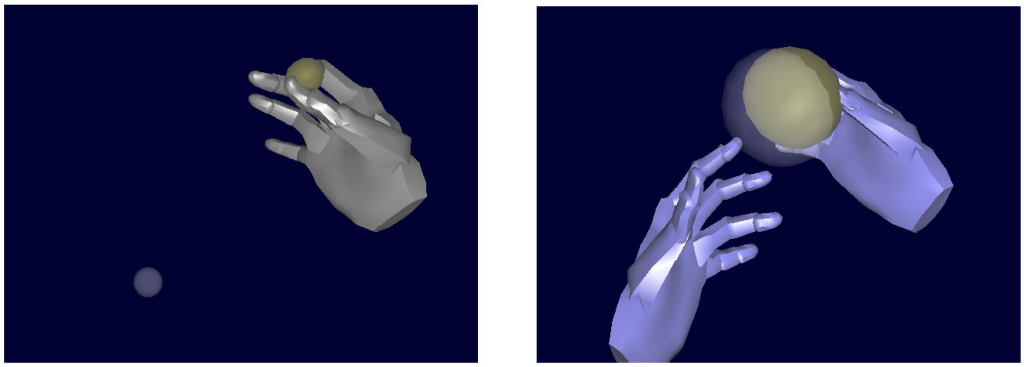
Figure 2.28: Scaled down movement with the PRISM technique (extracted from [42]).

rapidly. When users are focused on accurately moving an object to very specific locations, they normally slow their hand movements and focus more on being precise. PRISM increases the control/display ratio, which causes the cursor or object to move more slowly than the user's hand, thereby reducing the effect of hand instability and creating an offset between the object and the hand. Using PRISM, the user is always in complete control of the position of the object being manipulated (in contrast to gravity and snapping techniques). User evaluation results show faster performance and higher user preference for PRISM over a traditional direct approach.

The authors later extended the previous work by adding support for object rotation, which uses the angular speed of the hand [43] and which the authors concluded to be confusing to users. They also presented how their approach can be useful for faster object selection using a 3D cursor, either for out-of-reach objects using a smoothed ray-casting approach or for cluttered environments, such as the Worlds in Miniature approach [120].

Combining PRISM with the ray-casting-based approach HOMER [20], Wilkes et al. proposed Scaled HOMER [133], which uses velocity-based scaling to allow more precise manipulation at both near and far distances. It improved performance over HOMER in a wide variety of task conditions, primarily in those that require a high level of precision, object placement at a distance, or a large movement distance. Following Go-Go and PRISM studies, Auteri et al. [6] combined both techniques to increase precision for extended reach 3D manipulation. The solution starts by applying PRISM to the movement of the user's hand (base cursor) directly, which calculates a new cursor position (PRISM cursor) based on velocity-based scaling. Then, the distance that the PRISM cursor moved is amplified by the Go-Go distance-based heuristic. The combination of Go-Go and PRISM provided a number of improvements, particularly task completion success and fine-grained manipulation.

One- and two-handed control techniques for precise positioning of 3D virtual objects in IVEs were proposed by Noritaka Osawa [99]. This author proposed a position adjustment that consists of a scale factor for slowing hand movement, similar to PRISM [42], and a viewpoint adjustment that automatically approaches the viewpoint to the grabbed point such that the object being manipulated appears larger (Figure 2.29). To control the adjustments, two techniques are presented. The first uses only one hand and is based on its speed on the assumption that the user moves their hand slowly when they want to precisely manipulate an object. The other uses the distance between both hands. When the distance between them is small, the adjustments are activated. Through a user evaluation, the position and view-



(a) Normal stage (b) Precise stage with viewpoint adjustment

Figure 2.29: Viewpoint adjustment for increased precision, which causes the manipulated object to appear larger (extracted from [99]).

point adjustment methods showed improvements for small targets over a base scenario where this adjustments were disabled. Additionally, their results also showed that the two-handed control technique performed better than the one-handed technique.

Aguerreche et al. [1] introduced a 3D interaction technique called 3-Hand Manipulation for multi-user collaborative manipulation of 3D objects. The 3-Hand Manipulation technique relies on the use of three manipulation points that can be used simultaneously by three different hands of two or three users. The three translation motions of the manipulation points can fully determine the resulting 6-DOF motion of the manipulated object. When a hand is close enough to the object to manipulate, ray-casting from the hand provides an intersection point with the object. This point is called a manipulation point. A rubber band is drawn between a hand and its manipulation point to avoid ambiguity concerning its owner and to display the distance between the hand and the manipulation point. It is elastic, and its colour varies according to the distance between the hand and the manipulation point. The authors indicate that a possible solution for implementing their technique is to use three point-to-point constraints of a physics engine.

Inspired by the previous work, Nguyen et al. [96] proposed a widget consisting of four manipulation points attached to objects, called the 3-Point++ tool, which includes three handle points, forming a triangle, and their barycentre. With this widget, users can control and adjust the position of objects. By moving the manipulation points, the position and the orientation of the object are controlled. The barycentre can be used for approximate positioning to control the object directly without constraints, while the three handle points are used for precise positioning. For this purpose, the barycentre has 6 DOFs, while the three handle points have only 3 DOFs. If

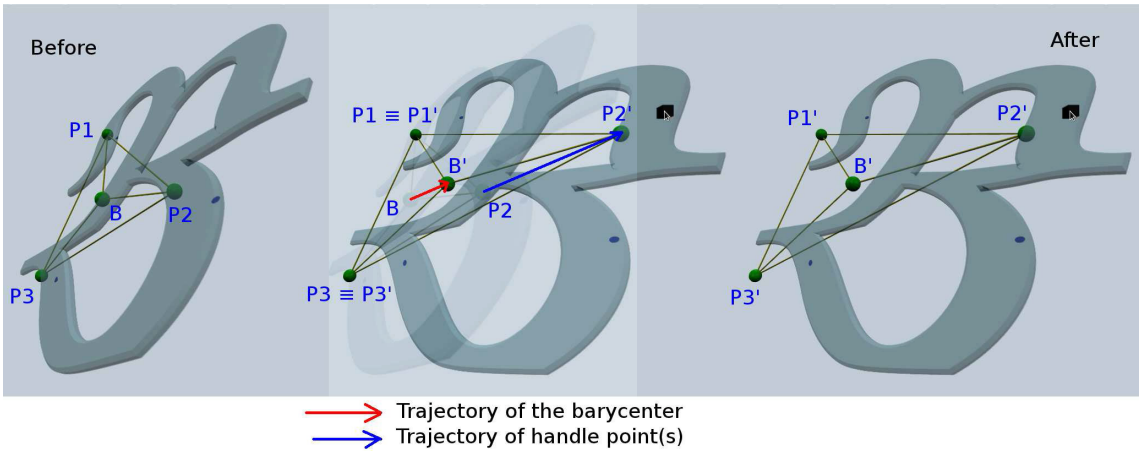


Figure 2.30: 3-Point++ tool: moving the handle point P2 causes the object to rotate around an axis created by the other two handle points P1 and P3 (extracted from [96]).

one handle point is manipulated, then the object is rotated around an axis created by the two other handle points, as shown in Figure 2.30. If two handle points are manipulated at the same time, then the object is rotated around the third handle point. An evaluation was conducted comparing the 3-Point++ tool with a well-known technique using a 3D cursor to control an object directly with 6 DOFs. The 3-Point++ technique had the worst results due to its complexity.

Extending their previous work, Nguyen et al. [97] presented the 7-Handle manipulation technique. This technique consists of a triangle-shaped widget with seven points, as depicted in Figure 2.31. Three points called first-level handles are the three vertices of the triangle, which act similarly to the 3-Point++ tool. The second-level handles are positioned at the midpoints of the three sides of the triangle and are used to control its two adjacent first-level handles. The last point, the third-level

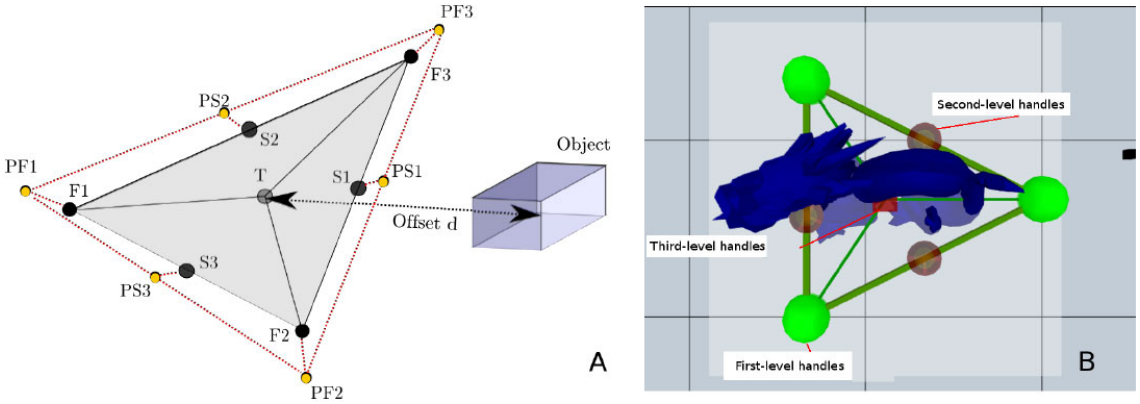


Figure 2.31: A: Set of seven points of the 7-Handle tool. B: Implementation of the 7-Handle tool (extracted from [97]).

handle, is positioned at the centroid of the three first-level handles and can be used as a direct manipulation tool with 6 DOFs. The results of a user evaluation showed that the 7-Handle technique is only better suited than the traditional direct 6-DOF approach for manipulating large objects (side larger than 1.5 metres).

2.2.3.7. Analyses and Comparisons

Several papers present comparisons of different methods with user experiments, attempting to derive interesting hints and design guidelines.

In [11], the use of 3D input is questioned after a comparison between mid-air manipulation with devices tracked in 6DOF and mouse-based methods on a placement task. Their experiment showed that, even with less DOF, the mouse was more efficient than the other devices. Its accuracy compensated the need to decompose tasks and it induced lower levels of stress.

Veit et al. [125] studied the influence of the integration and separation of DOFs in orientation tasks in semi-immersive VEs. For this purpose, they compared an indirect mid-air technique (IR - indirect rotation), in which users can grab a virtual manipulator (a cube) and orient it by rotating the hand, with another where users manipulate each rotation axis independently in a multi-touch surface (BPCR - bi-manual plane-constrained rotations). Using the IR technique, users are able to combine three axes of rotation into a single gesture. With the BCPR technique, a 3-DOF orientation task can be decomposed into three 1-DOF sub-tasks by manipulating one axis at a time. User evaluation results showed that participants were faster with BPCR and revealed that even when using IR, participants tended to decompose tasks.

Schultheis et al. [110] performed a comparison between mouse, wand and a two-handed interface for 3D virtual object and world manipulation through user evaluation, using both monoscopic and stereoscopic displays (although no discussion is provided regarding viewpoint correlation or co-location of users' hands and virtual imagery). The mouse interface resorted to manipulators (or widgets) for controlling translation and rotation angles for each axis. The wand behaved as a regular 6-DOF tracked device, allowing direct manipulation of the selected object. The two-handed approach is very similar to the Handle-Bar [116]. The two-handed interface out-performed the mouse and wand, and the wand out-performed the mouse, albeit requiring appropriate training. The authors stated that these results suggest that

well-designed many-DOF interfaces have an inherent advantage over 2-DOF input for fundamental 3D tasks.

Vuibert et al. [127] compared the performance of three mid-air interaction options using either a physical replica of the virtual object, a wand-like device or the user's fingers. For this purpose, they conducted a user evaluation with a docking task with 6 DOFs. As a baseline, they resorted to a mechanically constrained input device, the Phantom Omni. The authors found that the Phantom was the most accurate device for position and orientation, whereas the tangible mid-air interactions (wand and object's replica) were the fastest. Although the fingers did not outperform the Phantom in accuracy or speed, the difference between these two conditions was small. Moreover, subjects preferred the wand and fingers, while interaction with the replica was the least favoured.

Caputo and Giachetti [25] conducted an evaluation and comparison of four mid-air manipulation techniques using low-cost hand tracking sensors. The examined techniques ranging from direct to more indirect metaphors to study the effectiveness of DOF separation and hybrid solutions for different manipulation actions, such as translation and rotation. The usability of the methods was tested in an immersive VR environment with test subjects performing a simple docking task. The results showed better performance for all the techniques using a more indirect approach for rotation actions.

Moehring et al. [92] presented a study that compares finger-based interaction to controller-based interaction in a CAVE and in a HMD for exploration of car models. The authors focused on interaction tasks within reach of the users' arms and hands and explored several feedback methods, including visual, pressure-based tactile and vibrotactile feedback. The results suggest that controller-based interaction is often faster and more robust since the button-based selection provides very clear feedback on the start, stop and status of the interaction. However, finger-based interaction is preferred over controller-based interaction for the assessments of various functionalities in a car interior, as the abstract character of indirect metaphors leads to a loss of realism and therefore impairs the judgement of the car interior. Grasping feedback is a requirement to judge grasp status. It is not sufficient to simply have an object follow the user's hand motion once it is grasped. Although visual feedback alone is mostly sufficient for HMD applications, tactile feedback significantly improves interaction independent of the display system. Vibrational feedback is considerably stronger than pressure-based sensations but can quickly become annoying.

2.2.4. Discussion

We have seen that a variety of methods for virtual object manipulation, including rotation, translation and scaling, have been proposed, each with its own features. To analyze the manipulation techniques described in the previous sections and to identify trends and open challenges, we classified them according to the taxonomy presented in Fig 2.3. In addition to the taxonomy concepts, we also characterize each technique regarding environment properties presented in Figure 2.1. For techniques that were further developed, leading to a new and improved technique, we only consider their latest stage (e.g. touch-based Z-Technique [81] and DS3 [82], and mid-air 3-Point++ [96] and 7-Handle [97]).

The classifications of techniques are presented in Tables 2.1, 2.2, and 2.3. We use abbreviations for display properties (SC: screen constrained, SW: stereo window, and RR: reality replacement). We also identify whether it separates transformations and, for each transformation type, its mapping, the number of required contact points (CP), total transformation DOFs supported (TD) and the minimum explicitly simultaneously controlled DOFs (MD).

Additionally, regarding transformation separation, we indicate which transformations are grouped or set apart, e.g. {T,R,S} means that translation, rotation and scaling operations are applied simultaneously, whereas {T},{R},{S} indicates that it is possible to fully control all supported DOFs of each transformation separately from the other transformation. Although we do not go into further detail on which DOFs of each transformation are controlled together, we occasionally separate DOFs from a transformation to clarify how the separation is performed. For instance, {Txy},{T,Rz} means that translations on both x and y axes are applied simultaneously, but to also translate in the z axis, users must enable rotations around the same axis.

2.2.4.1. Desktop 3D Interfaces

As shown in Section 2.2.1, methods for 3D manipulation in desktop environments currently appear to be well established. Main editors and applications for 3D design generally exploit similar techniques and widgets, derived from techniques that are more than two decades old. Table 2.1 summarizes the surveyed works in 3D desktop manipulations. Naturally, and characteristic to desktop environments, all displayed imagery is screen constrained, and the tracking space is 2D separated.

Technique	Environment Properties		Transformations												
	Display Mapping	Tracking Space	Separation	Translation				Rotation				Scaling			
				Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Triad cursor[98]	SC	2D Separated	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	3	1	Remapped	1	3	1
Two Pointer[136]	SC	2D Separated	Partial: {T},{T,R},{R},{T,S}	Exact	1-2	3	2	Exact	2	3	1	Distance	2	3	1
Handle box[60]	SC	2D Separated	Total: {T},{R}	Hybrid	1	3	1	Remapped	1	1	1	No control			
Virtual Handles[32]	SC	2D Separated	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	3	1	Remapped	1	3	1
Arcball[111]	SC	2D Separated	Only {R}	No control				Remapped	1	3	1	No control			

Table 2.1: Classification of techniques for manipulating 3D virtual objects with desktop interfaces (SC: screen constrained, SW: stereoscopic window, CP: number of contact points required, TD: total transformation DOFs supported, MD: minimum explicitly simultaneously controlled DOFs).

The main challenge in desktop 3D virtual object manipulation is the mapping of the 2D input to 3D transformations. To overcome this challenge, either a multiple viewport approach and/or specific widgets are generally used. The multiple viewport approach uses different views of the virtual scene. These are orthogonal projections, and their view vector is coincident to a scene axis. Consequently, all interactions can be restricted to a single plane for each viewport, taking advantage of simpler 2D interactions and a more direct mapping between input and output. Widgets are a common alternative that allows interactions with unconstrained perspective projections of the 3D virtual environment. These consist of additional virtual objects that allow users to explicitly select specific transformations and axes to be applied onto the desired object.

As shown in the table, all surveyed mouse-based techniques allow single DOF control, with the exception of Two Pointer. While the former became the default for desktop manipulations, the latter became the basis for multi-touch interactions with two contact points, primarily for 2D manipulations, including the rubber band metaphor for scaling operations. Using multi-DOF devices rather than a mouse does not provide measurable advantages [11] and may be recommended only for specific applications or user categories.

2.2.4.2. 3D Manipulation on Interactive Surfaces

The main features of touch-based 3D manipulation interfaces proposed in the literature are summarized in Table 2.2. Analyzing this table, we observe that most techniques are conceived for co-located environments, as expected for multi-touch interfaces. It is also possible to observe that most perform DOF separation, decoupling not only transformations but also DOFs in each transformation supported. However, few explore scaling operations, and there are many remappings due to the dimensionality disparity between input and output.

The great advantage of touch-based interfaces is the possibility to interact with virtual content by directly touching it with ones' fingertips. This allows for more natural interactions, as physical manipulation metaphors can be employed, potentially reducing techniques' learning curves. Although 2D interaction has found easy-to-use *de facto* standards for multi-touch devices, adapting these standards to manipulate 3D objects is not trivial in that it requires mapping 2D input spaces to 3D virtual worlds. Attempting to create natural interactions, researchers initially proposed techniques for controlling several DOFs at the same time using exact map-

Technique	Environment Properties		Transformations											
	Display	Tracking	Separation	Translation				Rotation				Scaling			
	Mapping	Space		Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Sticky Fingers[56]	SC	2D Co-located	Partial: {Txy},{T,Rz},{T,R}	Exact	1-3	3	2	Exact	2-3	3	1	No control			
Screen-space[107]	SC	2D Co-located	Partial: {Txy},{T,Rz},{T,R}	Exact	1+	3	2	Exact	2+	3	1	No control			
DS3[82]	SC	2D Co-located	Total: {T},{R}	Hybrid	1-2	3	2	Hybrid	2+	3	2	No control			
Liu et al.[75]	SC	2D Co-located	Partial: {T,Rz},{Rxy}	Exact	2	3	2	Exact	2	3	1	No control			
tBox[30]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	1	1	Remapped	1	3	1
LTouchIt[88]	SC	2D Co-located	Total: {T},{R}	Exact	1	3	2	Remapped	2	3	1	No control			
Au et al.[5]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	2	3	1	Remapped	2	3	1	Distance	2	1	1
GimbalBox[16]	SC	2D Co-located	Total: {T},{R}	Remapped	1	3	2	Remapped	1-2	1	1	No control			
TouchSketch[135]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	2	3	1	Remapped	2	3	1	Distance	2	3	1
Triangle Cursor[121]	SW	2D Co-located	None: {T,R}	Hybrid	2	3	3	Exact	2	1	1	No control			
Toucheo[52]	SW	2D Separated	Partial: {T},{Txy,Rz,S},{Rxy},{S}	Hybrid	1-2	3	1	Hybrid	1-2	3	1	Hybrid	1-2	3	1
Indirect6[112]	SW	2D Separated	Total: {T},{R}	Hybrid	1-2	3	2	Remapped	2	3	1	No control			
Void Shadows[47]	SW	2D Co-located	Partial: {Txy},{T,Rz}	Exact	1-2	3	2	Exact	2	1	1	No control			

Table 2.2: Classification of techniques for manipulating 3D virtual objects with touch-based interfaces (SC: screen constrained, SW: stereoscopic window, CP: number of contact points required, TD: total transformation DOFs supported, MD: minimum explicitly simultaneously controlled DOFs).

pings [56, 107]. It has been shown that, when considering direct approaches that follow exact mappings, the numbers of input and output DOFs should be close. Thus, higher DOF transformations should be associated with a higher number of contact points.

The main issue with direct touch approaches for object manipulation is that, when controlling multiple DOFs simultaneously, unwanted transformations occur. To prevent this, reduction in the number of DOFs simultaneously controlled has been suggested [81, 82] and followed by several authors. By manipulating fewer DOFs at each moment, users have increased control over the outcome, which can also increase the efficiency of the manipulations. This DOF separation can consist of both separating different transformations and restricting a transformation to specific axes. For separating transformations, a common way to achieve this is by using a different number of touches for each transformation, e.g. one finger translates, two fingers rotate [82]. To identify a single transformation axis, the vector defined by two fingers can be used [5]. However, finding an adequate projection of such a 2D vector to the virtual scene to define a 3D vector might be challenging.

Alternatively, virtual widgets have been proven to be quite useful. Similar to mouse-based manipulations, other researchers turned to virtual widgets to clearly and unambiguously select the transformation and axis [30, 88, 16]. Evaluation results suggest that these can improve users' performance. Widgets can show all the manipulations available for an object, or a set of transformations according to a specific axis through user sketching. They ease the process of remembering how to perform restricted transformations, generally by touching on specific handles.

When using touch to interact with stereoscopic imagery above tabletops, different challenges arise since directly touching on a displayed object might disrupt depth's illusion and/or suffer from parallax issues. Proposed solutions follow indirect approaches, either by touching outside the object, typically resorting to some type of widget to remap users' actions, or by using separated interaction spaces through additional touch-enabled surfaces. The only technique that allows full 9-DOF manipulations in such settings resorts to virtual widgets [52].

2.2.4.3. Mid-air Interactions

Once again, we applied our taxonomy to classify techniques for mid-air manipulation, as reported in Table 2.3. From this table, we can conclude that most techniques, although being developed for several types of displays and tracking solutions, resort

to exact mappings due to the naturality offered by spatial input. Thus, very few explore transformation separation, and only partially. Even less support DOF separation within a transformation type. Again, as in touch-based manipulations, few explore scaling operations, and those that do only support uniform scaling.

Exact mapping between tracked hand/device and virtual object has been followed in most current mid-air approaches for 3D virtual object manipulation [73, 58, 3, 116, 128]. This is the most natural approach as it mimics physical interactions, and studies have shown that it is well suited for coarse transformations. Almost none of the proposed mid-air techniques with exact mappings has separation of transformations. The only exception is Air-TRS [3], as transformations are enabled with different hands, being possible to translate without performing any rotation. However, it does not allow performing rotations with one hand without having the other hand engaged in translations.

Having realized that human accuracy is limited, occasionally aggravated by input devices' resolution, efforts have been conducted to alleviate this issue. Possible approaches to improve manipulations' accuracy scale down users' hand motions [42, 43], or move the viewpoint closer to the object being manipulated [99]. However, it has been shown that scaling down hand motions is only appealing for translations, as scaled rotations generally confuse users, severely decreasing overall performance. Another way to tackle the lack of precision in mid-air is DOF separation. DOF separation has the potential to provide better accuracy and prevent unwanted transformations, and, as it is common in mouse- and touch-based techniques, it can be achieved through virtual widgets, for instance. Approaches based on virtual widgets have been proposed to limit simultaneous transformations in mid-air, but these do not provide promising results: 3-Point++ [96] performs worse than direct manipulation with 6 DOFs, and 7-Handle [97] is only suited for very large objects. In this dissertation, we will propose more familiar virtual widgets that offer transformation separation and single DOF manipulation, using common reference frames. We believe they might yield better results than those proposed by previous techniques.

Despite only translation and rotation are required for positioning tasks, scaling is often grouped together with those transformations in specialized 3D software. However, in the reviewed mid-air techniques scaling is very much disregarded. Some touch approaches that resort to widgets allow for single DOF scaling [30, 5, 135, 52], but mid-air techniques that offer scaling capabilities only perform uniform scaling [116, 29, 3].

Technique	Environment Properties			Transformations											
	Display Type	Tracking Space	Hands/DOF Tracked	Separation	Translation				Rotation				Scaling		
					Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Simple Virtual Hand [73]	RR	3D Co-located	1/6	None: {T,R}	Exact	1	3	3	Exact	1	3	3	No control			
In the Air[58]	SC	3D Separated	1/4	None: {T,R}	Exact	1	3	3	Exact	1	1	1	No control			
Air-TRS[3]	SW	3D Co-located	2/3	Partial: {T},{T,R,S}	Exact	1-2	3	3	Exact	2	3	3	Distance	2	1	1
VHGM[65]	RR	3D Co-located	1/6	None: {T,R}	Exact	1	3	3	Exact	1	3	3	No control			
Handle Bar[116]	SC	3D Separated	2/3	Partial: {T,Ryz},{Rx}	Exact	2	3	3	Hybrid	2	3	1	Distance	2	1	1
Spindle+Wheel[29]	SW	3D Separated	2/6	None: {T,R}	Exact	2	3	3	Hybrid	2	3	3	Distance	2	1	1
Crank Handle[18]	SW	3D Separated	1/3	Total: {T},{R}	Exact	1	3	3	Remapped	1	3	1	No control			
Grasping Object[18]	SW	3D Separated	1/3	Partial: {T},{T,R}	Exact	1	3	3	Remapped	1	3	3	No control			
PRISM[43]	RR	3D Co-located	1/6	None: {T,R}	Scaled N:1	1	3	3	Scaled N:1	1	3	3	No control			
Viewpoint Adjustment[99]	RR	3D Co-located	2/6	None: {T,R}	Exact	1	3	3	No control				No control			
7 Handle[97]	RR	3D Co-located	2/3	None: {T,R}	Remapped	1-2	3	3	Remapped	1-2	3	1	No control			
Go-Go[103]	RR	3D Co-located	1/6	None: {T,R}	Scaled 1:N	1	3	3	Exact	1	3	3	No control			
HOMER[20, 133]	RR	3D Co-located	1/6	None: {T,R}	Scaled 1:N	1	3	3	Exact	1	3	3	No control			
Worlds in Miniature[120]	RR	3D Co-located	2/6	None: {T,R}	Remapped	1	3	3	Remapped	1	3	3	No control			
Voodoo Dolls[102]	RR	3D Co-located	2/6	None: {T,R}	Remapped	1	3	3	Remapped	1	3	3	No control			

Table 2.3: Classification of techniques for manipulating 3D virtual objects in mid-air (SC: screen constrained, SW: stereoscopic window, RR: reality replacement, CP: number of contact points required, TD: total transformation DOFs supported, MD: minimum explicitly simultaneously controlled DOFs).

Another result of exact mappings is that they may require additional movement of the user, either physical or virtual. When designing interactions for large environments, it is useful to explore techniques for extending users' reach. The first obstacle in manipulating out-of-reach objects is the selection of the virtual object. Despite the various studies on object selection, objects further away than room-sized distances still pose some challenges, as it is impractical to interact with hand-sized copies of very large environments [120] and ray-casting approaches can be highly inaccurate for longer distances. To circumvent this, a selection volume can be used instead of a ray [74, 41, 119, 84], but it requires progressive refinement strategies as multiple objects can be intersected.

Progressive refinement techniques often favor closer objects either rearranging them [26, 70] or automatically refining selection based on proximity [41, 48], which might not be feasible for selecting far-away objects. Also, these category of techniques employ menus [70] or metaphors for diminishing the field of view (FOV) with zoom operations [9], and were developed for non-immersive and non-stereoscopic scenarios. Thus, they may not be suited for VR as menus might disrupt immersion and "small FOVs may lead to cybersickness" [73]. In this matter, there are few works that employ progressive refinement in IVEs, but they resort to additional interactive surfaces which may also lead to reduced immersion [86]. The Disambiguation Canvas solves this by representing the touches on a mobile phone screen in the IVE [37] to disambiguate between objects, but they do not support selection of far-away objects. After a successful selection, manipulation can be achieved either using scaled mappings [20, 133] or handheld copies of the object [102]. In our work, we will introduce a way to employ an iterative progressive refinement strategy within IVEs to perform selections, and thus allow interactions with out-of-reach objects.

Although techniques that move away from direct manipulations are less natural, they can avoid unwanted side effects of replicating the physical world exactly, such as poor mid-air accuracy and limited reach. They have the capability to provide users with enhanced abilities that may improve performance and usability [23].

2.3. Exploratory Work

During this PhD research, we also conducted some work focused on topics marginal to the thesis core. The respective research contributed with valuable knowledge and

experience, both with 3D interaction techniques and interactive setups. Below we present these works, as well as the scientific publications they originated.

2.3.1. Interactive Stereoscopic Visualization of Architectural Models

In the context of the Alberti Digital research project, we developed an interactive installation to didactically explore the classic architectural treatise written by Leon Battista Alberti, De Re Aedificatoria, as generative design systems, namely shape grammars. It allows users to interactively explore such architectonical knowledge in both appealing and informal ways, by enabling them to visualize and manipulate in real-time different design solutions. This installation was available to the general public in several museum exhibitions in Portugal related to Alberti's treatise, specifically at the Science Museum of the University of Coimbra, the National Museum of Natural History and Science, and the Faculty of Architecture of the University of Lisbon.

Using a stereoscopic tabletop, with a touch sensitive surface, users can perform virtual visits to existing buildings that follow Alberti's treatise and to create custom ones according to rules from the treatise. Also, using a non-invasive spatial tracking based on depth cameras, visitors can point at specific model parts in mid-air, using the forefinger. The system, after detecting a collision between the finger end and the model, highlights that model part and shows an informative label, containing that part designation. Manipulation of the virtual model can be achieved using several touches on the interactive surface. We developed a finger-cluster interaction method, which allows users to move, rotate and uniformly scale the models. While a single touch only allow translation, with two or more fingers all the transformations can be applied simultaneously. Mid-air interaction, due to its inconsistent accuracy, was only used for pointing and not for manipulation, and was eventually left out in the latter exhibitions to simplify the setup.

Publications:

- Bruno Figueiredo, Eduardo Castro e Costa, Bruno Araújo, Fernando Fonseca, **Daniel Mendes**, Joaquim Jorge, José Pinto Duarte (2014). *Interactive Tabletops for Architectural Visualization.* 32nd Conference in Education and

Research in Computer Aided Architectural Design in Europe (eCAADe '14), pp. 585-592.

- Bruno Araújo, **Daniel Mendes**, Fernando Fonseca, Alfredo Ferreira, Joaquim Jorge, Eduardo Costa, Bruno Figueiredo, Filipe Coutinho, José Pinto Duarte (2014). *Interactive Stereoscopic Visualization of Alberti Architectural Models.* Joelho: Revista de Cultura Arquitectónica N. 05, pp. 52-56. DOI: 10.14195/1647-8681_5_5

2.3.2. 3D Object Retrieval with Speech and Immersive Visualization

Focusing on natural interactions for searching in large collections of 3D digital objects, we presented a novel interface for 3D object retrieval in immersive virtual environments. As a proof of concept, we developed a prototype using the context of LEGO blocks, combining speech for query specification with stereoscopic visualization and mid-air input for result navigation. We explored different modes for displaying results in front and around the user, and proposed a new one based on their best aspects. We also experimented with two different input devices: the Leap Motion and the SpacePoint Fusion. The latter was chosen due to its increased accuracy and users' preference, even though it does not offer a hands-free interaction. With the development of this work, it became noticeable the difficulties associated to pointing at distant objects. Moreover, our prototype only allows the placement of a retrieved block by pointing at the desired position, since this was outside the scope of this research. An improved mid-air manipulation technique, supporting full 6 DOF translation and rotation, would contribute to better interactions.

Publications:

- Pedro Pascoal, **Daniel Mendes**, Diogo Henriques, Isabel Trancoso, Alfredo Ferreira (2015). *LS3D: LEGO Search Combining Speech and Stereoscopic 3D.* International Journal of Creative Interfaces and Computer Graphics (IJCICG) 6(2), pp. 18-36. DOI: 10.4018/IJCICG.2015070102

- Diogo Henriques, Isabel Trancoso, **Daniel Mendes**, Alfredo Ferreira (2014). *Verbal description of LEGO blocks.* 15th Annual Conference of the Interna-

tional Speech Communication Association (INTERSPEECH '14), pp. 1618-1622.

- Diogo Henriques, **Daniel Mendes**, Pedro Pascoal, Isabel Trancoso, Alfredo Ferreira (2014). *Evaluation of Immersive Visualization Techniques for 3D Object Retrieval*. IEEE Symposium on 3D User Interfaces (3DUI '14), pp. 145-146. DOI: 10.1109/3DUI.2014.6798862

2.3.3. Collaborative Virtual Environments for 3D Design Review

Within the CEDAR research project, we dealt with offshore engineering projects that often engage geographically distributed highly-specialized engineering teams. They require both improved productivity and reduced risks when reviewing 3D designs of deep-water oil and gas platforms, due to project costs. We developed novel approaches to support the discussion and decision making in 3D design reviewing tasks in virtual environments. Also, current video-conference and collaborative technologies add unproductive layers of protocol to the flow of communication between participants, rendering the interactions far from seamless. We introduced Remote Proxemics, an extension of proxemics aimed at bringing the syntax of co-located proximal interactions to virtual meetings, and Eery Space, a shared virtual locus that results from merging multiple remote areas where meeting participants are located side-by-side as if they shared the same physical location. Eery Space promotes collaborative content creation and seamless mediation of communication channels based on virtual proximity. This work helped us have a better grasp of the importance of interacting effectively and efficiently with 3D content in engineering projects, and made us take the first steps in developing what would become the user tracker based on depth cameras used in Chapters 4 and 5.

Publications:

- Maurício Sousa, **Daniel Mendes**, Daniel Medeiros, Alfredo Ferreira, João Pereira, Joaquim Jorge (2016). *Remote Proxemics*. In Collaboration Meets Interactive Spaces, pp. 47-73. DOI: 10.1007/978-3-319-45853-3_4

- Maurício Sousa, **Daniel Mendes**, Alfredo Ferreira, João Pereira, Joaquim Jorge (2015). *Eery Space: Facilitating Virtual Meetings Through Remote Prox-*

emics. Human-Computer Interaction (INTERACT '15), pp. 622-629. DOI: 10.1007/978-3-319-22698-9_43

- João Guerreiro, Daniel Medeiros, **Daniel Mendes**, Maurício Sousa, Joaquim Jorge, Alberto Raposo, Ismael Santos (2014). *Beyond Post-It: Structured Multimedia Annotations for Collaborative VEs*. In Proceedings of the 24th International Conference on Artificial Reality and Telexistence and the 19th Eurographics Symposium on Virtual Environments (ICAT-EGVE '14), pp. 55-62. DOI: 10.2312/ve.20141365

- **Daniel Mendes**, Maurício Sousa, Alfredo Ferreira, Joaquim Jorge (2014). *ThumbCam: Returning to single touch interactions to explore 3D virtual environments*. In Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14), pp. 403-408. DOI: 10.1145/2669485.2669554

2.3.4. Illustration of Layer-cake Models on and above a Touch Surface

In this work, we explored the potential of mid-air hand gestures to geological modeling. Typically, illustrating and visualizing 3D geological concepts are performed by sketching in 2D mediums, which may limit drawing performance of initial concepts. We developed a spatial interaction prototype to enable rapid modeling, editing, and exploration of 3D layer-cake objects, using a setup similar to that used in Section 2.3.1. User interactions are acquired with mid-air tracking and touch screen technologies, and a semi-immersive virtual environment is provided with a stereoscopic tabletop. The novelty consists of performing expeditious modeling of coarse geological features with only a limited set of hand gestures, which proved to be more efficient when compared to a WIMP counterpart.

Publications:

- Daniel Simões Lopes, **Daniel Mendes**, Maurício Sousa, Joaquim Jorge (2016). *Expeditious illustration of layer-cake models on and above a tactile surface*. Computers & Geosciences Volume 90 Part A, pp. 1-9. DOI: 10.1016/j.cageo.2016.02.009

2.3.5. Virtual Reality for Radiologists

Whenever radiologists are ill-positioned with respect to the display or visualize images under improper light and luminance conditions, serious diagnostic errors can occur. In the IT-MEDEX research project, we developed a prototype to show that virtual reality can assist radio-diagnostics by considerably diminishing or cancel out the effects of unsuitable ambient conditions. It combines immersive head-mounted displays with interactive surfaces to support professional radiologists in analyzing medical images and formulating diagnostics. In order to deal with resistance to non-typical medical practices and procedures, our approach relies on rendering 3D medical data which is placed to float above a virtual desk in VR.

To interact with the medical volume, we applied DOF separation to some extent. We resorted to bimanual interactions where each hand can simultaneously perform separate and independent actions. Similar to the work of Section 2.3.1, we allow people to touch the desk the way they feel more comfortable with by treating all fingers of the same hand as a single contact point. With the left hand, medical professionals can either adjust the image brightness or navigate through the available slices. After enabling one of these actions, it stays locked until the hand is lifted. Right hand gestures are reserved for scaling and rotation transformations. Changing the volume's scale can be achieved using a pinch gesture. Pitch rotation is done by moving the hand forwards and backwards, and yaw rotation by rotating the hand. Yaw rotation is snapped to the nearest anatomical plane upon releasing, with 90^{o} changes.

Publications:

- Paden Shorey, Audrey Girouard, Sang Ho Yoon, Yunbo Zhang, Ke Huo, Karthik Ramani, Mauricio Sousa, **Daniel Mendes**, Soraia Paulo, Nuno Matela, Joaquim Jorge, Daniel Simões Lopes, Dirk Wenig, Johannes Schöning, Alex Olwal, Mathias Oben, Rainer Malaka (2017). *Demo hour.* ACM Interactions Magazine Volume 24 Issue 6, pp. 8-11. DOI: 10.1145/3143318
- Maurício Sousa, **Daniel Mendes**, Soraia Paulo, Nuno Matela, Joaquim Jorge, Daniel Simões Lopes (2017). *VRRRoom: Virtual Reality for Radiologists in the Reading Room.* In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17), pp. 4057-4062. DOI: 10.1145/3025453.3025566

2.3.6. Mid-Air Modeling with Boolean Operations in VR

CSG (constructive solid geometry) is a powerful enabler for more complex modeling tasks, allowing to create complex objects from simple ones via Boolean operations. In the context of the TECTON 3D research project, we conceived two new mid-air interaction techniques to achieve Boolean operations between two objects while immersed in VR. One is based on direct object manipulation via gestures while the other uses menus, and we conducted a preliminary evaluation of these techniques. Additionally, Head-Mounted Displays occlude the real self, and make it difficult for users to be aware of their relationship to the virtual environment. To account for self-representation, we compared full-body avatar against an iconic cursor depiction of users' hands. The results of this research highlighted once again the limitations associated to the mid-air object manipulations' accuracy.

Publications:

- **Daniel Mendes**, Daniel Medeiros, Maurício Sousa, Ricardo Ferreira, Alberto Raposo, Alfredo Ferreira, Joaquim Jorge (2017). *Mid-Air Modeling with Boolean Operations in VR*. IEEE Symposium on 3D User Interfaces (3DUI '17), pp. 154-157. DOI: 10.1109/3DUI.2017.7893332

2.4. Chapter Summary

In this chapter, we explained some key concepts related to object manipulation within virtual environments, to facilitate the proper understanding of the work presented. We described: characteristics of input and output devices and technologies used to interact with VE, ranging from the mouse to mid-air tracking and from the traditional desktop screen to immersive head-mounted displays; the main stages of object manipulation, namely selection and transformation; and other transformation related concepts, such as degrees-of-freedom and mappings.

We also surveyed the state-of-the-art regarding virtual object manipulation with different interaction paradigms. We covered 3D desktop interfaces, manipulations on interactive surfaces and mid-air interactions, both with traditional screens and

immersive displays. We saw that, in order to solve the required mapping between the 2D input and the 3D transformations, DOF separation has been a highly followed approach for mouse and touch based interfaces, with proven results. While for mid-air interactions the input/output mapping is simpler, the attainable precision is quite limited. Some manipulation techniques have been proposed to mitigate this, mostly for immersive environments, but there is still significant amounts of research before mid-air approaches are able to rival with traditional interfaces. For instance, exploring the impact of DOF separation in mid-air, in contradiction to the natural metaphors commonly used, remains scarcely explored. Additionally, the lack of accuracy in mid-air also affects interactions with object outside users' arms reach, since even small jitters can significantly erode performance in pointing actions with increased distance. An unexplored way to tackle this matter in VR is to employ iterative progressive refinement strategies, providing that they are suitably adapted.

Lastly, we presented several projects carried out during this thesis, with topics marginal to its focus, where we explored several hardware configurations and different application scenarios, such as architecture, engineering and medicine. They confirmed the need for this thesis research and contributed for the development of tools, experience and knowledge that made the work presented in the following chapters possible.

3
Initial Assessments

Many computer applications require virtual three dimensional object manipulations, such as architectural modeling, virtual model exploration, engineering component design and assembly, among others. Because of this, 3D manipulation has been the focus of intense research. Also, to improve both the visualization and spacial perception of the three dimensional content, several researchers explored interactions using stereoscopic environments. These allow users hands to be co-located in mid-air with the virtual objects, offering natural interactions. While 2D interaction with multi-touch devices has clearly found *de facto* standards, the same cannot be said about mid-air interactions. Due to different requirements and/or tracking capabilities, different approaches for manipulating virtual objects in mid-air have been proposed.

In this chapter we compare the most common mid-air manipulation techniques, checking their performance and user preferences through a user evaluation, in Section 3.1. Next, in Section 3.2, we conduct another evaluation to assess the performance differences of the two best techniques in immersive and semi-immersive VEs. These evaluations contributed for the definition of the path followed and the baselines used in the ensuing studies.

3.1. Mid-Air Manipulation Techniques

Stereoscopic displays allows users to manipulate three dimensional entities as if they were co-located in mid-air with their hands and body. Naturally, the most common approach to perform such manipulations is to use a Simple Virtual Hand metaphor [73], as it mimics interactions with physical objects, offering a direct mapping of a hand tracked in 6 DOF. Nonetheless, such approach has its limitations: a full 6 DOF tracking might not always be available, and other non natural operations may be required, such as the scaling transformation that is often grouped with translation and rotation. Therefore, other techniques have been explored [3].

Interacting with stereoscopic environments using interactive surfaces has also been the subject of previous research [52, 121, 10]. However, these are restricted to a two-dimensional space and cannot offer direct interactions above the table. Other researchers [116] proposed solutions to interact within a three-dimensional space, but do not combine them with stereoscopic systems, using them only as a more powerful, yet still indirect, cursor.

In this first study, we aimed to understand which approaches to 3D virtual object manipulations, based on the literature, are best suited to interact within a SIVE, using a stereoscopic tabletop. We implemented five different techniques, four mid-air and one multi-touch, which was used as a baseline. We performed a user evaluation to compare the techniques and to understand users' preference regarding 3D virtual object manipulations in mid-air.

We start by describing the implemented object manipulation techniques, followed by the presentation of the method, tasks and prototype used, as well as the participants that took part in the evaluation. Then we report and discuss the attained results, regarding both task performance and user preferences. Lastly, we present the major lessons took from this study.

3.1.1. Implemented Techniques

Based on the existing literature, we chose five different techniques to assess which are best suited to manipulate virtual objects placed in mid-air. Four of these use mid-air spatial input, for both direct and indirect interactions, and one is touch-based. All implemented techniques provide 7 DOF (three for translation, three for rotation, and one for uniform scaling).

3.1.1.1. 6-DOF Hand

To mimic interactions with physical objects, as closely as possible, 6-DOF Hand follows the Simple Virtual Hand approach [73], using all the 6 DOF tracking information of the users' dominant hand (3 DOF for position and 3 DOF for orientation). With this technique, users can grab the desired object directly with one hand, typically the dominant one, and all hand movements are directly applied to the object, as depicted in Figure 3.1. Dragging the object in space moves it in three dimensions, and wrist rotations controls object rotation.

Unlike the Simple Virtual Hand, 6-DOF Hand also supports uniform scaling. Grabbing somewhere in the space outside the object with the non-dominant hand, and varying its distance to the dominant hand, scales the object using the nowadays common rubber stretching metaphor [136]. Increasing the distance makes the object larger and, analogously, the object becomes smaller when the distance decreases. The initially grabbed point in the object will remain the center of all transformations, during the entire manipulation, until the object is released.

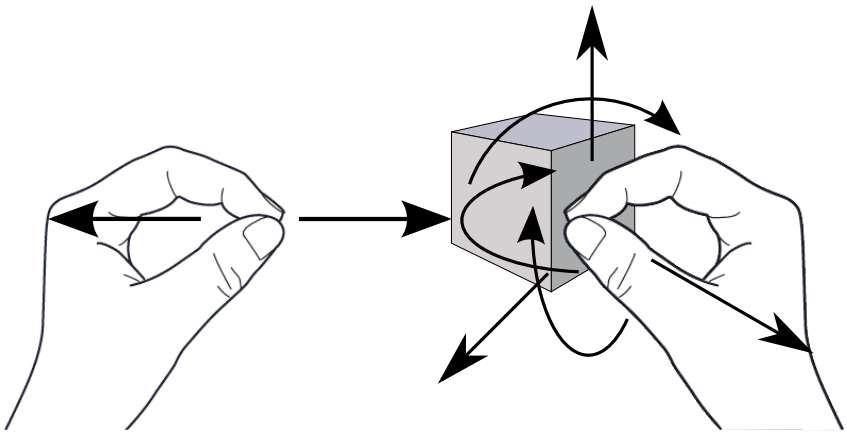


Figure 3.1: The 6-DOF Hand technique. The hand that grabs the object directly controls its translation and rotation. The distance between both grabbed hands uniformly scales the object.

3.1.1.2. Handle-Bar

Following the work of Song et al. [116], we included the Handle-bar metaphor in our assessment. This approach simulates a physical bimanual handle-bar, commonly used, for example, to steer a bicycle. This technique requires only 3 DOF positional tracking of each and hand, and uses the middle point (MP) between both hands to manipulate virtual objects (Figure 3.2).

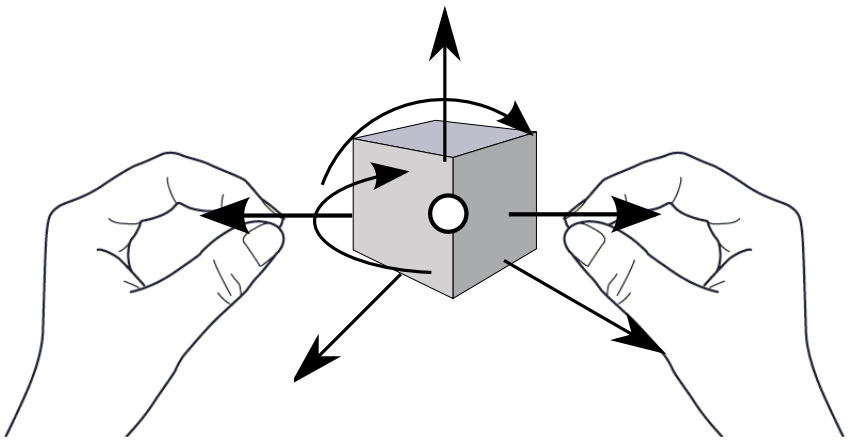


Figure 3.2: The Handle-Bar technique. The middle point of both hands is used to manipulate the object, reacting as if the user was holding a bar placed across the object. The distance between both hands scales the object.

After performing a grab gesture with both hands, an object intersected by the MP can be manipulated. The object is translated by moving both hands in the same direction, and rotated by moving the hands in different directions. Changing the distance between hands evenly scales the object. Similar to the 6-DOF Hand, all three transformations can be applied at the same time, but here the MP is the center of the transformations. The object is released after both hands performed a release gesture.

3.1.1.3. Air-TRS

Using only 3 DOF positional tracking, we can also extend the two-point Translate-Rotate-Scale (TRS) to the third dimension, as proposed by Araujo et al. [3]. We use hands' position in a similar fashion to the two-dimensional Two-Point Rotation and Translation with scaling [57], considering the third dimension as well, as illustrated in Figure 3.3.

The hand that is used to initially gab the object moves it by dragging. The other hand, after grabbing somewhere in space, not necessarily in the object, allows users to control rotation and scaling transformations. The rotation is defined by the variation in the position of one hand relatively to the other. For scaling, the variation in the distance between both hands is used. The transformations are centered in the point where the first hand grabbed the object, and are active until the hand that grabbed the object releases it.

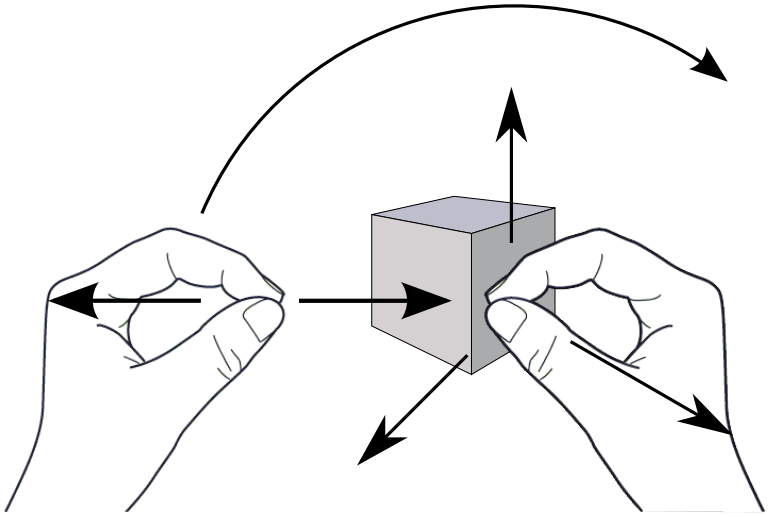


Figure 3.3: The Air TRS technique. The first hand grabs and moves the object. The movement of the second hand relatively to the first defines rotation and scaling transformations.

3.1.1.4. 3-DOF Hand

DOF separation attained good results for manipulating virtual objects in mouse- and touch-based interfaces, preventing undesired transformations [95]. Following this approach, 3-DOF Hand is an attempt to separate translation and rotation while keeping a direct mapping of wrist rotations.

In this technique, we divided translations and rotations by both hands (Figure 3.4). After grabbing the object with one hand, users can only perform 3 DOF translations by dragging it. Rotations are achieved by rotating the wrist of the other hand after

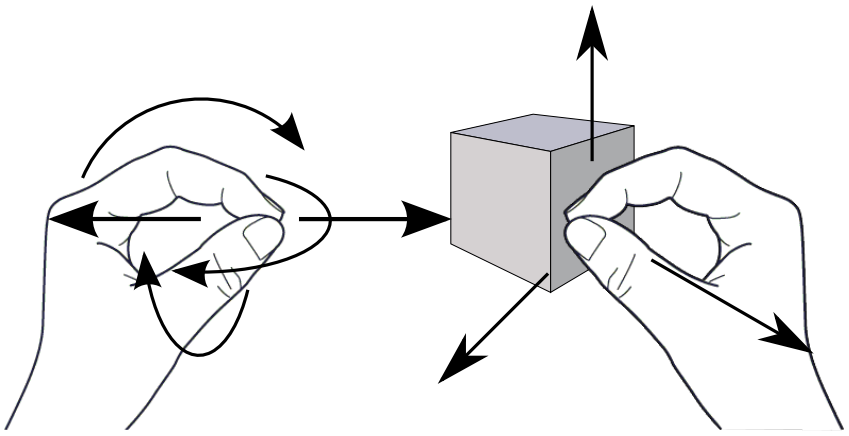


Figure 3.4: The 3-DOF Hand technique. The hand that grabs the object directly controls its translation. The rotations of the other hand define the object orientation. The distance between both hands scales the object.

performing a grabbing gesture somewhere in space, while keeping the object grabbed with the first hand. Similarly to the previous techniques, varying the distance between hands uniformly scales the object. The point grabbed in the object by the first hand remains as the center of all transformations until it is released.

3.1.1.5. Touch TRS + Widgets

Although only allowing indirect manipulations of virtual objects in the three-dimensional space above the surface, multi-touch is nowadays a commonly used input method, present in our everyday life. Adding to this familiarity, it is also an input method highly adequate to a greater DOF separation, due to the low DOF provided by each touch. Therefore, this technique acts as a baseline in our evaluation.

This touch technique can be seen as simplified version of the interactions used in Toucheo [52], also combining the 2D TRS algorithm with widgets to perform manipulations. This implementation provides DOF separation, allowing users to translate a virtual object in a plane parallel to the surface, by touching below it with one finger and dragging. While this touch is active, three widgets appear to the left or to the right of the touch, depending on which hand the finger corresponds to. By using a second finger outside of any widget, users can perform rotations around a vertical axis and scale the object. If the second touch is on one of the three widgets, users will be able to rotate around one of the two axes parallel to the surface

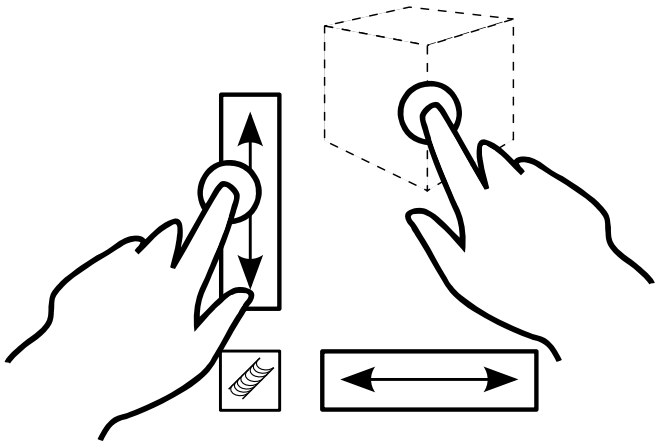


Figure 3.5: The Touch TRS + Widgets technique. One touch bellow the object enables widget visibility and moves the object. A second touch outside the widgets apply the TRS algorithm (translation and yaw rotation). The widgets offer height manipulation, roll and pitch rotations.

following a rod metaphor [52], or to modify object's height similarly to the balloon metaphor [10].

3.1.2. User Evaluation

To evaluate the techniques described above, we carried out a user evaluation. We aimed at identifying which were the faster and easier to use, as well as which were preferred by participants. To accomplish that, we tested our techniques in a practical task scenario, using a setup comprised of a stereoscopic tabletop and depth cameras. For each technique, participants had to perform a set of three different tasks with increasing difficulty.

3.1.2.1. Procedure

The experiment was performed in our laboratory, with a controlled environment. Every evaluation session for each participant followed the same protocol, starting with a short briefing explaining the experiment they were about to perform. Each technique was evaluated in partial random order, to ensure that all methods were experienced in every position at least once between all participants, to avoid biased results due to participants becoming acquainted to tasks and more used to the technology.

For each technique, a brief demonstration video showing how to use it was played. After the video, subjects had two minutes to explore the technique in a training scenario, as depicted in Figure 3.6. After participants felt comfortable using the technique, we asked them to perform a set of tasks. For each task, we allowed a maximum of five repetitions within a limited time in order to prevent sessions from lasting too long. Participants filled out questionnaires to classify each technique after completing the corresponding tasks, and an additional questionnaire was used for profiling purposes. On average, each session took around 60 minutes to complete.

3.1.2.2. Tasks

We devised three tasks for participants to execute in this evaluation. These were docking tasks as introduced by Zhai and Milgram [137], and followed a wooden toy

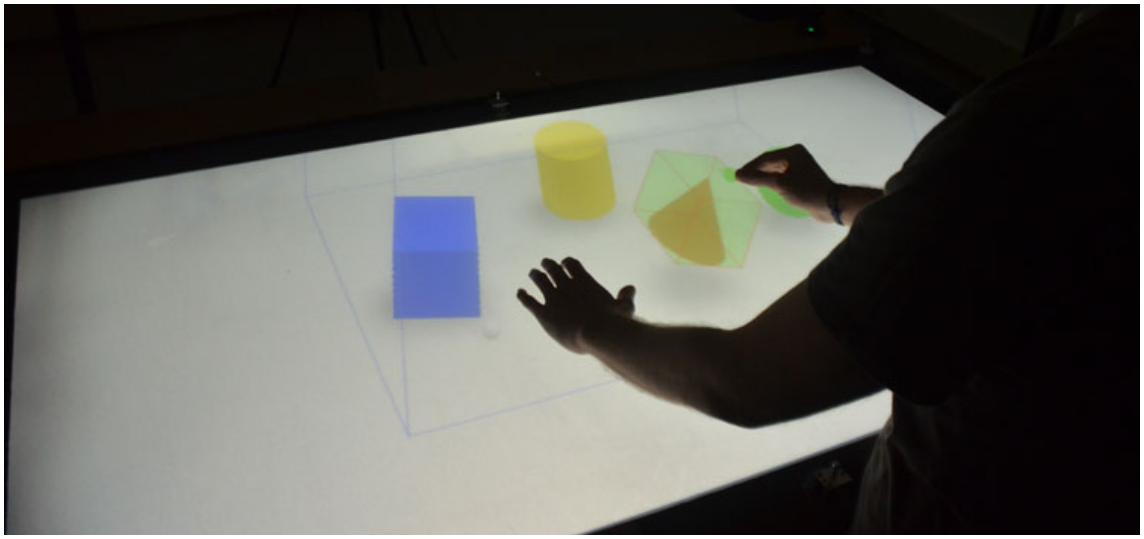


Figure 3.6: Participant manipulating objects in our training scenario.

metaphor as the peg-in-hole task used by Martinet et al. [82]. They require subjects to fit an object inside a hole with a similar shape in another object. To provide incremental difficulty between tasks, we started with an easy task, requiring only one kind of transformations, followed by an intermediate one, which needed two distinct transformations, and ended up with the most complex, that required all three transformations to be applied.

In all tasks, when a participant fulfilled the completion requirements, the object became white and locked in the current position. We chose to introduce this restriction to avoid tracker problems and user frustration when releasing an object that was already correctly placed. After the participant released the object, it was moved to a random position within a predefined set and always at the same distance to the target, in order to create different completion paths.

The first task consisted only in translations on a two-dimensional plane parallel to the surface, requiring neither height translation, nor rotation or scaling. In this task we asked the participants to put a sphere inside a box with a hole as many times as they could within one minute, as depicted in Figure 3.7.A. Although the object could be rotated, this was not considered for task completion. For this task alone, scaling was deactivated in all techniques. Therefore, the only requirement was the sphere to be placed within ten millimeters of the target position.

The second task required translations in all three axes and also scaling transformations, but it did not require any rotations. The scenario for this task consisted in a stylized torus, and a box with a torus-shaped hole in the front face (Figure 3.7.B). Each participant had to fit the torus inside that hole as many times as they could

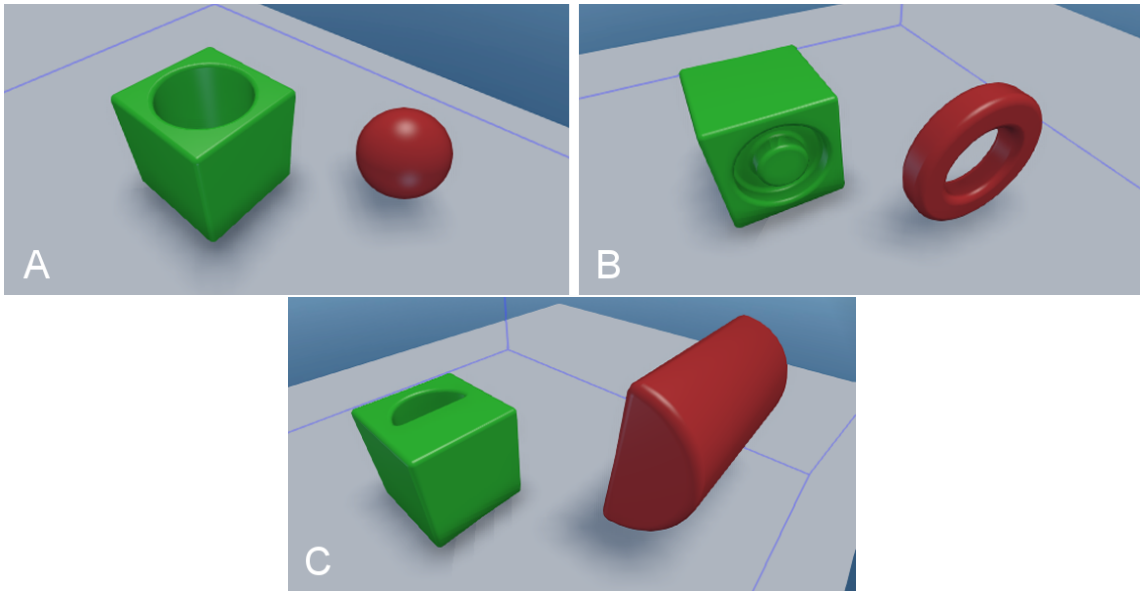


Figure 3.7: User evaluation tasks. First task (A) consists in fitting a sphere inside the hole of the box. Second task (B) consists in fitting a stylized torus inside the hole on the front box face. Third task (C) consists on fitting the semi-cylinder inside the box hole.

within two minutes. In this task, even though no rotations needed to be performed, the object needed to have an orientation according to the hole. So, the error tolerated for a task to be considered completed was ten millimeters for position, ten degrees for orientation, and ten percent for scale.

For the third task, we asked participants not only to translate and scale, but also to rotate the object. The scenario, illustrated in Figure 3.7.C, consisted in a semi-cylinder and a box having a hole with the same shape. As for the previous tasks, we asked participants to fit the cylinder inside the cube, keeping in mind that scaling, rotation and position were equally important. It was possible to fit the object in two different orientations and both were accepted. The error threshold was set at ten millimeters, 15 degrees and ten percent for position, orientation and scale, respectively. We asked subjects to try and complete this task as many times as they could within a three minute interval.

3.1.2.3. Setup and Prototype

To be able to compare the techniques, we developed a prototype where we implemented them. It can create the illusion of virtual objects placed in mid-air through a stereoscopic visualization with head tracking, and allows hands-free interactions using depth cameras.

Virtual Environment

Our prototype's virtual environment was built on top of the G3D engine [83]. It showed objects' shadows, but it supported neither gravity nor object collisions. Participants could only grab the movable object. When users intersected an object with their hands, a red wired box appeared around that object. To grab it, users could then perform a pinch gesture, as shown in Figure 3.8, and the wired box changed to a solid transparent green to provide visual feedback of a successful grab. By opening their hands, users could release the object and the bounding box disappeared.

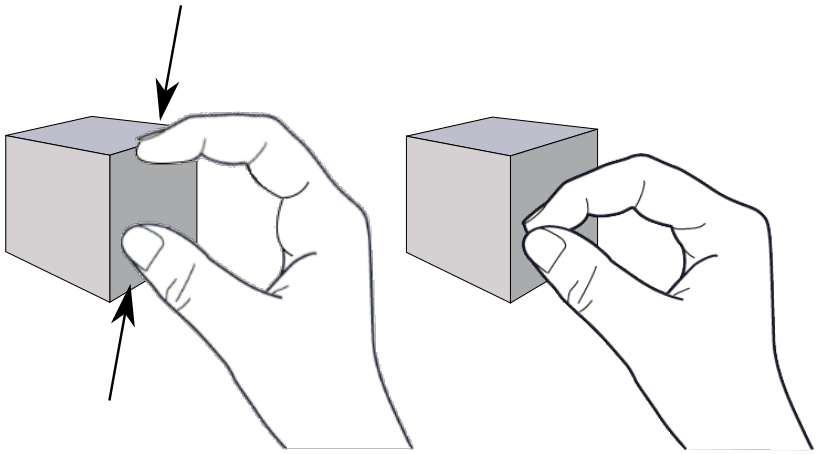


Figure 3.8: Pinch gesture in mid-air interaction techniques to grab an object.

Hardware Setup

Our prototype's hardware setup used affordable and non-intrusive solutions. It improved on a traditional multi-touch tabletop, with a stereo enabled display, adding depth cameras for head and hand tracking, as illustrated in Figure 3.9. The tabletop uses a Laser Light Plane technique to detect user touches. A NVidia Quadro K5000 equipped computer, paired with a stereo capable 120Hz HD Ready projector and NVidia 3D Vision glasses, enabled the stereoscopic render and assured that the correct image reached each eye. We used a depth camera (Microsoft Kinect v1), placed behind the tabletop, to track users' skeleton. Then we use the tracked position of users' head to generate the corresponding visualization frustum.

To explore mid-air interactions, we used an additional depth camera (also Microsoft Kinect v1) placed above the surface and pointing downwards. This camera tracked users' hands, capturing the corresponding point cloud and allowing their reconstruction. For this purpose, we used the 3Gear SDK, unfortunately no longer available.

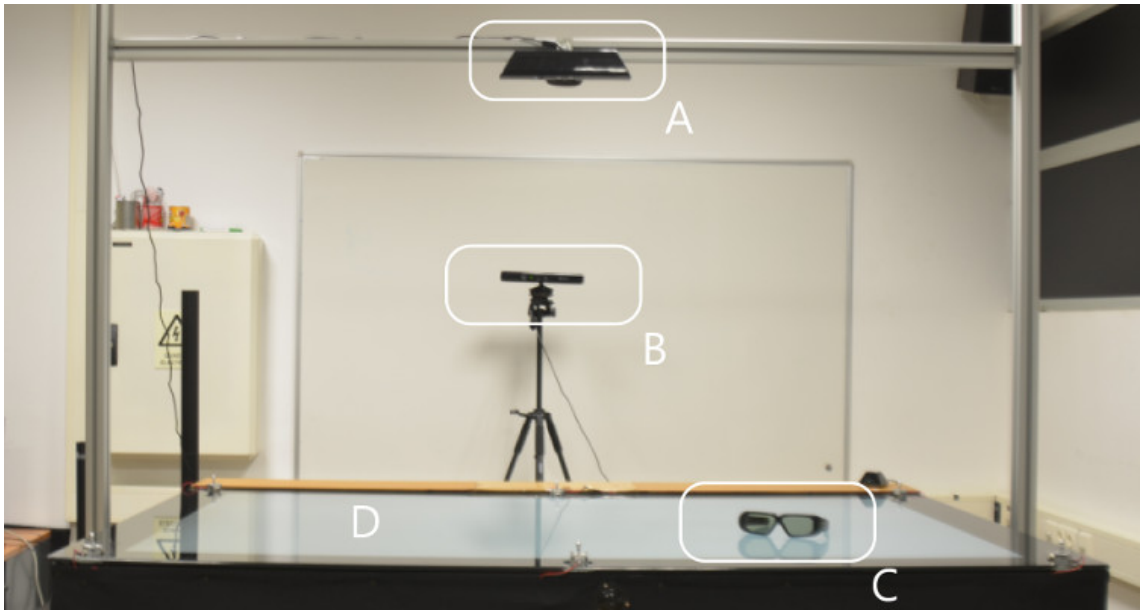


Figure 3.9: Our stereoscopic multi-touch tabletop (D) enhanced with depth cameras for non-intrusive tracking of head (B) and hands (A). Active shutter glasses (C) ensure the correct image for each eye.

By doing so, we could get not only the hand positions, but also their orientation and pose, without forcing the user to wear specific hardware. This information allowed us to explore spacial interaction techniques above the table in a non-intrusive manner, making the setup more immediate. Since the camera was placed above the surface and pointing downwards and it needed to capture the fingers used to pinch, participants were advised to avoid wrist rotations that would finish with such fingers pointing down and being occluded by the hand, as this could result in a non released object and lead to unwanted movements.

3.1.2.4. Participants

Twelve subjects participated in the user evaluation, eleven males and one female. Participants' ages ranged from 19 to 35 years old (median between 25 and 35) and all held at least a bachelor's degree. Only two participants did not own a touch device, such as a smartphone or tablet. Furthermore, only five had previous experience with stereoscopic visualization devices, and only three had experience with 3D modeling applications.

3.1.3. Results and Discussion

To measure task performance, we logged the time spent for each participant to complete each task for every interaction. After completing all tasks for each technique, participants answered a brief questionnaire regarding different interaction aspects. A final questionnaire also included queries to profile participants. Furthermore, we registered relevant participant actions and comments both during and after the experiment.

3.1.3.1. Task Performance

For each task we measured the time taken to complete it. Since users repeated the task several times for each technique, we averaged their times, in order to obtain a more accurate value per user. Results were tested for normality using the Shapiro-Wilk test and for sphericity with the Mauchly's test. Then, since all data was normally distributed, we used a repeated measures ANOVA to assess statistically

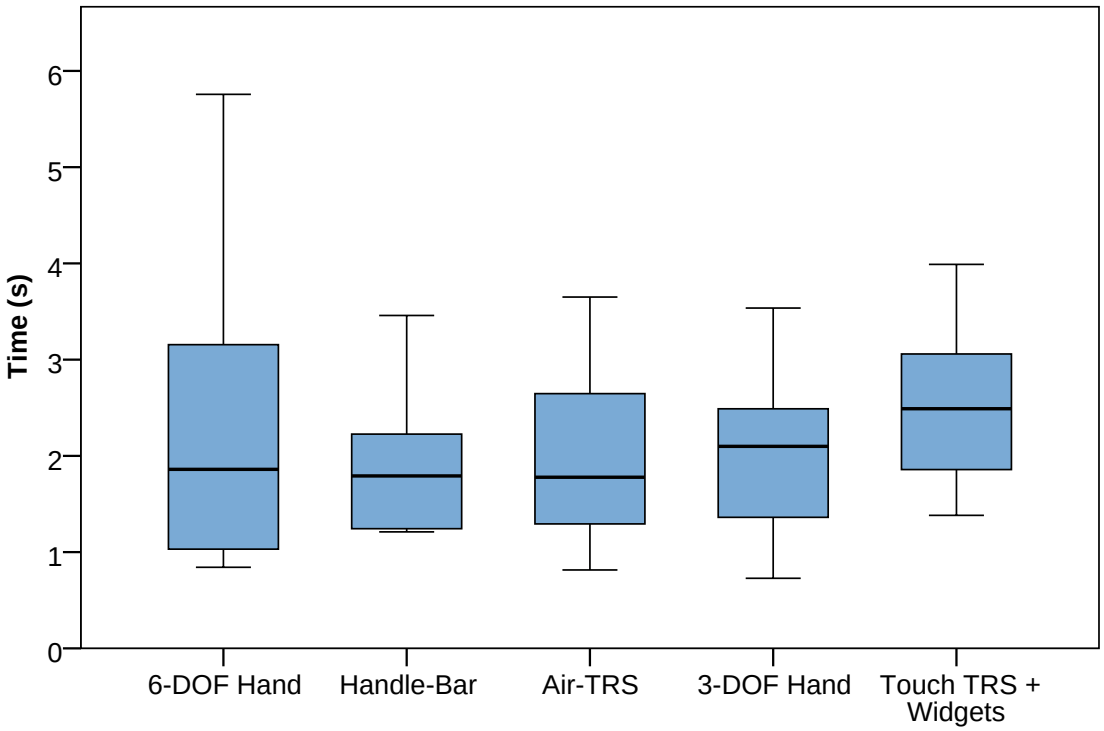


Figure 3.10: Time to complete the first task using the five techniques. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

significant differences. When significant differences were found, post-hoc tests used the Bonferroni correction (presented sig. values are corrected).

For the first task, the completion times are illustrated in the chart of the Figure 3.10 (average times: 6-DOF Hand 2.27s, Handle-Bar 1.88s, Air-TRS 1.95s, 3-DOF Hand 2.0s, Touch TRS + Widgets 2.53s). The repeated measures ANOVA found no statistically significant differences. This absence of significant differences among the techniques can be justified by the fact that only translation was needed to complete this task and all evaluated techniques are similar with regard to translation. Indeed, if we consider translation with one hand, 6-DOF Hand, Air-TRS and 3-DOF Hand are exactly the same.

Regarding the second task, the times required by the participants to complete the task are plotted in Figure 3.11 (average times: 6-DOF Hand 9.39s, Handle-Bar 3.40s, Air-TRS 6.99s, 3-DOF Hand 9.73s, Touch TRS + Widgets 18.1s). Applying the same repeated measures ANOVA for the results obtained in this task, we found statistically significant differences ($F(2.302, 20.722) = 16.757$; $p < 0.0005$). Notably, using the Handle-Bar significantly reduced the completion times (6-DOF Hand $p = 0.035$, Air-TRS $p = 0.002$, 3-DOF Hand $p = 0.004$, Touch TRS + Wid-

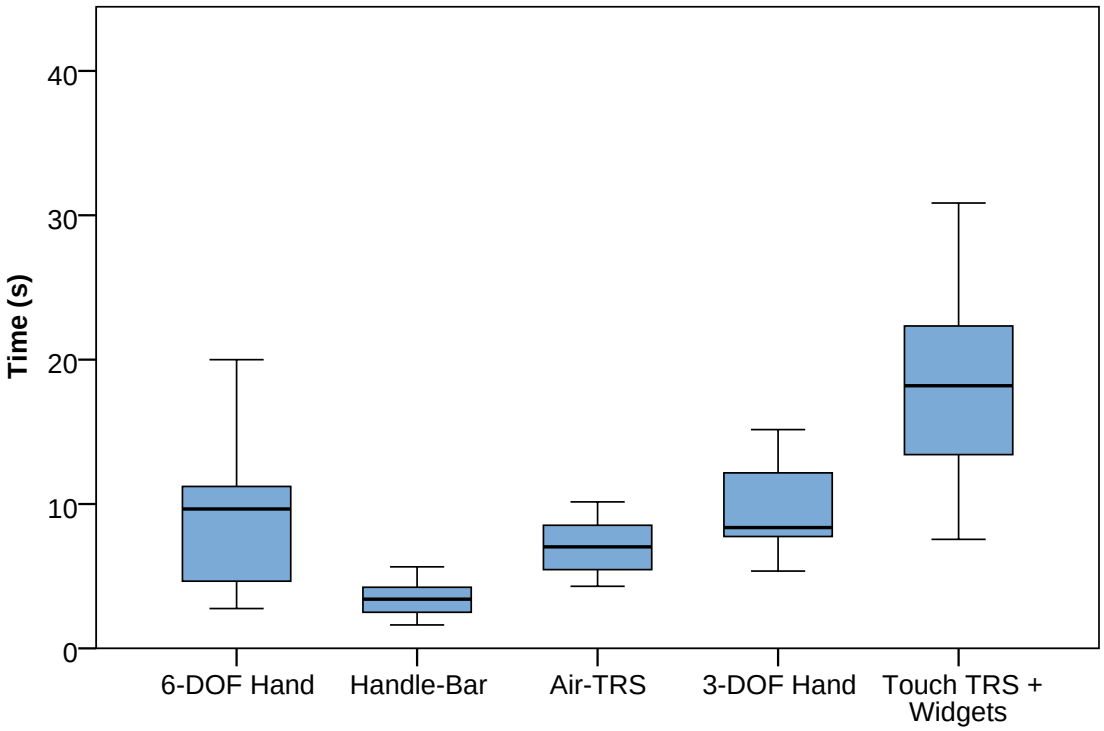


Figure 3.11: Time to complete the second task using the five techniques. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

gets p = 0.001). Touch TRS + Widgets was also significantly slower than Air-TRS (p = 0.002). One of the possible reasons why Handle-Bar was the fastest technique is that since the task required only translation and scaling, after grabbing the object, participants already have both hands in position to change the scale while moving. Moreover, since this technique only uses hand position, discarding wrist rotations, there were less unwanted rotations. The Touch TRS + Widget approach requires constantly changing between two touch TRS and different widgets, which led to the slower completion times. Again, the non-existence of significant differences between 6-DOF Hand, Air-TRS nad 3-DOF Hand relates to their resemblance, even for tasks requiring both translation and scaling.

Analyzing the completion times for the third task, which are depicted in Figure 3.12 (average times: 6-DOF Hand 20.6s, Handle-Bar 23.7s, Air-TRS 63.3s, 3-DOF Hand 46.9s, Touch TRS + Widgets 51.6s), statistically significant differences were found ($F(1.678, 11.748) = 8.697$, $p = 0.006$). Both 6-DOF Hand and Handle-Bar stood out, with significantly lesser completion times than 3-DOF Hand (6-DOF Hand $p = 0.046$, Handle-Bar $p = 0.037$) and much less dispersion than the others. We believe that the Handle-Bar was one of the fastest techniques, because it was the only method that did not lead to occlusions caused by subjects' hands, besides

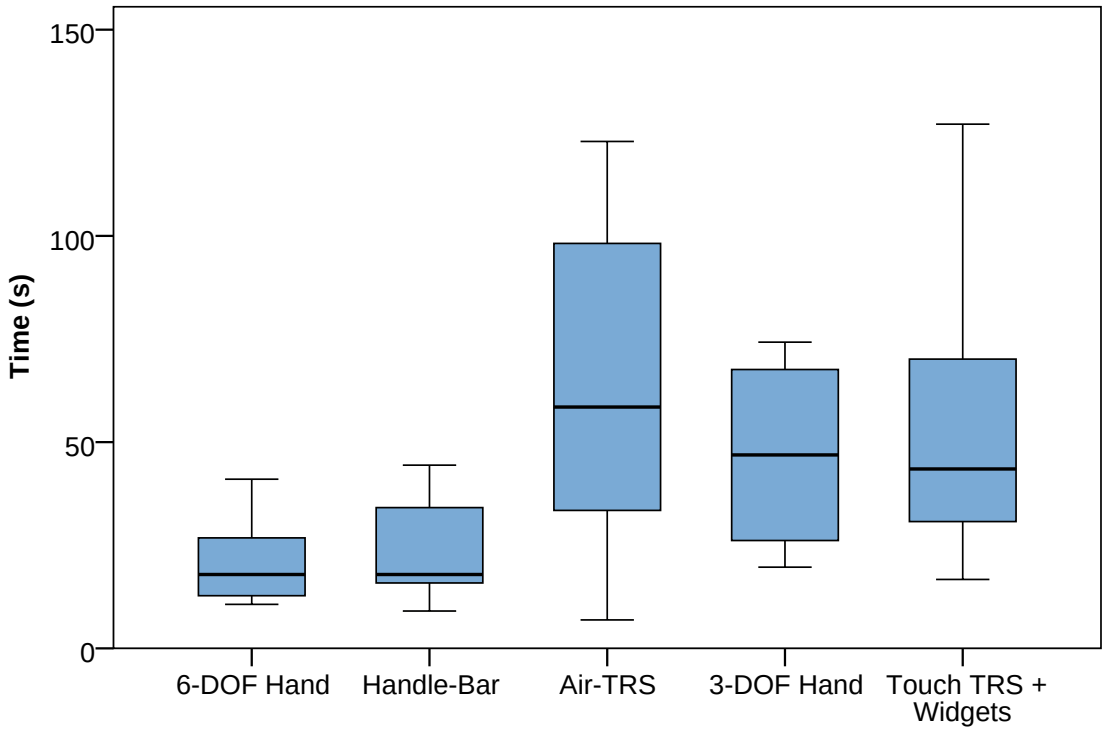


Figure 3.12: Time to complete the third task using the five techniques. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

that based on touch. This allowed a better perception of the object relative to its target position and orientation. The 6-DOF Hand mimics interactions with physical objects, which appeared more natural to participants.

3.1.3.2. User Preferences

In the questionnaires, we asked participants to classify each technique regarding five different criteria using a five point Likert scale (1 - very bad, 5 - very good). The answers are depicted in Table 3.1. We used the Friedman non-parametric test and the Wilcoxon Signed-Rank post-hoc test with the Bonferroni correction to assess whether differences were statistically significant. In what concerns translation of objects, the five techniques presented no statistically significant differences in terms of preference. On the other hand, statistically significant differences were found for rotation ($\chi^2(4) = 22.302$, $p < 0.0005$), scaling ($\chi^2(4) = 14.262$, $p = 0.007$), fluidity ($\chi^2(4) = 15.786$, $p = 0.003$) and fun ($\chi^2(4) = 27.588$, $p < 0.0005$).

For rotation, participants strongly agreed that 3-DOF Hand is more difficult to use than 6-DOF Hand and Touch TRS + Widgets ($Z = -2.965$, $p = 0.03$ and $Z = -2.976$, $p = 0.03$). The possible reason for participants to dislike 3-DOF Hand for rotations may reside in that rotating an object using the opposite hand that is grabbing it is not natural and may require some experience.

In terms of scaling objects, participants strongly agreed that the Handle-Bar was easier to use than 3-DOF Hand ($Z = -2.913$, $p = 0.04$). This can be a result of the Handle-Bar allowing scaling transformations right after starting an interaction with an object, which makes this transformation faster to be accessed, combined with the

	6-DOF Hand	3-DOF Hand	Handle-Bar	Air-TRS	Touch TRS + Widgets
Translation	4,5 (1)	4 (1)	4 (2)	4 (1)	4 (2)
Rotation *	4 (2)	2 (2)	3 (2)	3 (2)	4 (2)
Scaling *	4,5 (1)	3,5 (2)	5 (1)	4 (2)	4 (0)
Fluidity *	4 (1)	3 (1)	4 (1)	4 (2)	3,5 (3)
Fun *	5 (1)	2 (1)	4 (1)	4 (2)	4 (1)

Table 3.1: Participants preference for each technique regarding different criteria (median, interquartile range). * indicates statistical significance.

cognitive overload associated to the second hand with 3-DOF Hand, that requires the use of that hand not only to control rotations in a way that participants found difficult but also to perform scaling.

Regarding interaction fluidity, participants strongly agreed that 6-DOF Hand is superior to 3-DOF Hand ($Z = -2.994$, $p = 0.03$). These opinions can be explained because 6-DOF Hand mimics interactions with physical objects, which makes actions seem more natural, and a single grabbing action enables both translation and rotation. Finally, considering the fun factor, subjects strongly agreed that 3-DOF Hand was less amusing than 6-DOF Hand ($Z = 2.992$, $p = 0.03$), Handle-Bar ($Z = 2.877$, $p = 0.04$) and Air-TRS ($Z = 2.850$, $p = 0.04$). Also, they considered 6-DOF Hand more entertaining than the Handle-Bar ($Z = 2.887$, $p = 0.04$). We think that the directness and easiness of 6-DOF Hand mimicking physical interactions, as well as the high unnaturality of 3-DOF Hand, explain this result.

3.1.3.3. Observations

During the entire experiment we observed and registered everything relevant that participants said and did. While we implemented all techniques to support bimanual operation, we did not require any specific hand to be used first when grabbing objects. This enabled subjects to choose the most convenient hand to start the manipulation. While we assumed users to choose the dominant hand first, as suggested by Guiard asymmetric bi-manual model [51], we observed that participants changed their preference depending on which hand was closer to the object. The same was verified when each hand controlled a different degree of freedom e.g. for the 3-DOF technique. To rotate objects, some participants used the dominant hand to rotate the object rather than using it to control its position in space independently of their handedness.

For each task, we recorded the time spent performing it using either a single hand or both hands. Figure 3.13 presents the percentage distribution for all users in the third task which involved greater variability in transformations (translation, rotation and scale). Depending on the technique, we noticed a strong dependence on the DOF distribution. Obviously, we do not discuss the Handle-Bar method since it requires both hands. Regarding Touch TRS + Widgets, we could see that users spent almost the same time using one or two hands. This may be because participants kept the object selected with one finger while figuring out on the next transformation needed to fulfill the requirements. Both 3-DOF Hand and Air-TRS require one

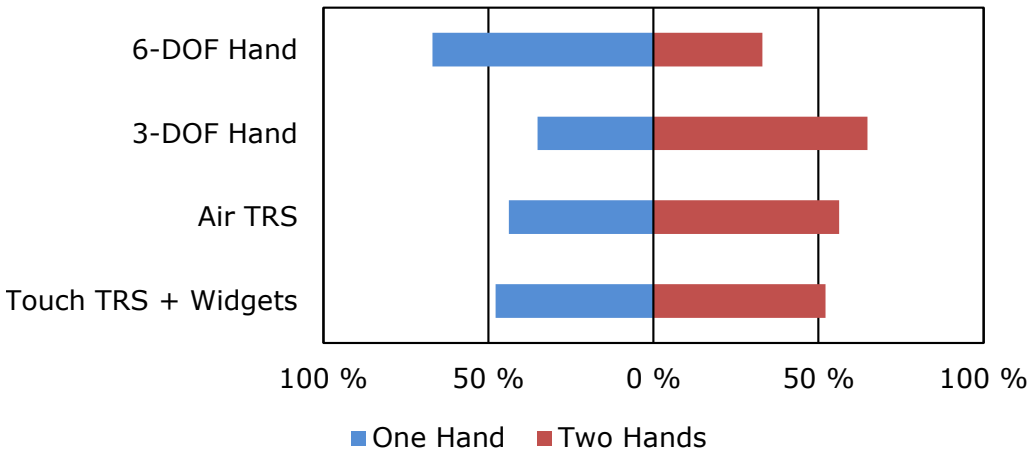


Figure 3.13: Time distribution of one- and two-handed object manipulations in the third task of our evaluation.

hand to translate and two hands to rotate and scale the object, thus showing a similar time distribution between one and two hand manipulations. However, the required hand gesture for rotation is different for each technique and the 3-DOF Hand induced participants to spend more time trying to achieve the right object rotation. The 6-DOF Hand is the approach that needed less time using two hands, because it closely mimics physical direct interactions, requiring only one hand to simultaneously execute both translation and rotation. The second hand is only needed to perform scale transformations which scarcely happens.

When rotating objects, subjects using 3-DOF Hand and 6-DOF Hand methods complained about poor tracker fidelity, which created unwanted rotations and increased the difficulty of tasks. These problems were also noticeable when releasing objects. Another interesting fact about translations regards Touch TRS + Widgets, where participants always started using the balloon widget to the wrong side of the final position they wanted to acquire. Still concerning interactions, many participants felt an initial difficulty when trying to rotate objects using either Air-TRS or the Handle-Bar. Additionally, many participants complained that mid-air interactions that operate directly on objects (6-DOF Hand, 3-DOF Hand and Air-TRS) suffer from having the selected object occluded by their hand, which does not happen with the Handle-Bar and Touch TRS + Widgets. Despite finding them tedious, subjects agreed that touch interactions are easy to use. Some participants also indicated that they missed a widget to perform TRS transformations on the surface.

3.1.4. Lessons Learned

In this study, we compared five different techniques to manipulate three-dimensional virtual objects in mid-air, based on the literature. Four of these techniques used mid-air interactions and one resorted to a multi-touch surface. Using a stereoscopic multi-touch tabletop setup with non-invasive user head and hand tracking, we evaluated each implemented technique via three tasks of increasing difficulty to try and find out which method was favored in terms of object manipulation. To this end, we registered task completion time, as well as participant preferences, meaningful actions, and comments throughout every session. Finally, we performed a quantitative and qualitative analysis.

Participants agreed that the 6-DOF Hand approach was more natural to use, since it reproduces direct interactions with physical objects. Results also showed that the Handle-Bar was sometimes faster than the 6-DOF Hand. Additionally, we observed that approaches that directly grabbed the manipulated object created unwanted occlusions, a consequence of stereoscopic displays already identified in the literature [24, 27]. This did not affect the Handle-Bar. Also, the hand tracking based on Microsoft Kinect depth cameras, although offering walk-up-and-use mid-air interactions, created some difficulties related to pose and orientation detection.

The main conclusion of this evaluation is that, concerning virtual objects lying in space, mid-air manipulations that have a greater resemblance to interactions in the physical world appeal more to users. In fact, Feng et al. [39] more recently conducted an evaluation similar to ours, but they used a different setup with held devices. Rather than co-locating users' hands with the stereoscopic imagery, they used a fish tank stereoscopic visualization and implemented manipulation techniques with a fixed offset. Similarities in their results in relation to ours lead to a tentative guideline: if satisfying each individual user's preference is of high importance to the interface designer, provide users with the option of Handle-Bar or 6-DOF Hand derived methods; otherwise, use an approach similar to the 6-DOF Hand.

3.2. Immersive versus Semi-Imersive Virtual Environments

Due to its relevance, manipulation of 3D virtual objects in mid-air has been subject of research and several different approaches have been proposed. However, not all were proposed for the same environments, as specific requirements or hardware limitations may exist. For instance, some use position and orientation tracking information of users' hands to offer direct 6 DOF manipulations, while others can only make use of positional information. Additionally, some were developed for systems that co-locate users' hands and the virtual content, while others are used for indirect interactions.

With this study we wanted to assess which technique is best suited to manipulate 3D objects in mid-air, with co-located interactions, both in IVE and SIVE. We also compared the performance and preferences for these two kinds of VE. We used a stereoscopic tabletop setup for the SIVE, and a HMD for the IVE, both paired with a depth camera and two hand-held controllers. For the manipulation techniques, we selected the two that positively stood out in the previous study. To compare the environments and the techniques, we conducted a user evaluation.

We begin by detailing the VE and the techniques of our study. Then we describe the user evaluation, including the method, tasks, prototypes, apparatus and participants. This is followed by the results' analysis, divided in task performance and user preferences, and by the discussion of the lessons learned.

3.2.1. Virtual Environments Tested

Mid-air manipulations of 3D virtual objects can be performed in a variety of configurations. We are interested in those that can co-locate users' hands and the virtual objects in the same space, allowing for direct interactions in a somewhat immersive environment. One of the key features required for this to be supported is the use of stereoscopy. In our previous study, we resorted to a stereoscopic tabletop to achieve a SIVE, due to hardware availability. Now, we wanted to find out how it compares to a fully immersive VE, using a HMD, for manipulation purposes.

3.2.1.1. Semi-Immersive Virtual Environment

For the SIVE, we used a tabletop for displaying the virtual content through a stereoscopic window approach with negative parallax, visible in Figure 3.14. As such, users can still see the their own body as well as the physical space surrounding them. By knowing where the user's head is, and considering the interpupillary distance, it is possible to create a different visualization frustum for each eye, thus enabling a stereoscopic visualization with a high viewpoint correlation [49]. This can create the illusion of virtual objects to be laying in the space above the tabletop.

However, and as previously stated, such approach can cause undesired objects' occlusions. Since users' hands are always in front of the screen, they will occlude the imagery even when the virtual objects should be on top of them. Also, this kind of environments have limitations on the available interaction space, as it is restricted to the space above the sceen.

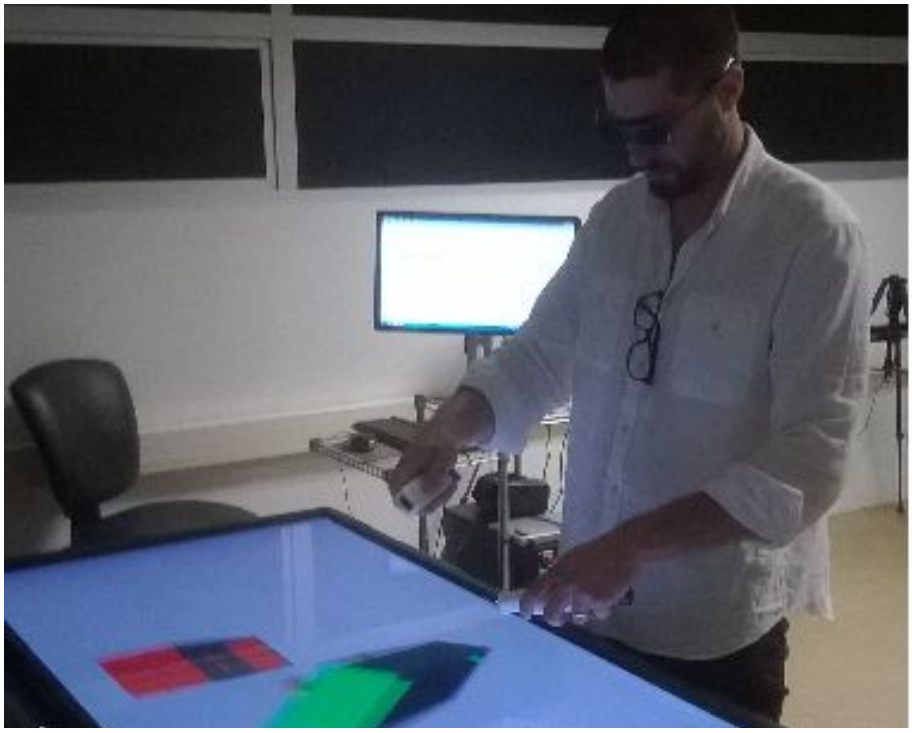


Figure 3.14: Participant experimenting the SIVE.

3.2.1.2. Immersive Virtual Environment

IVEs have the property of totally replacing users' surrounding physical space by the virtual environment, placing them inside the virtual world and offering an increased

Figure 3.15: Participant using the IVE.

immersion. Therefore, users can freely look in every direction and potentially have a greater interaction space than SIVEs. With head's position and orientation tracking, it is possible to have a total viewpoint correlation [49]. We resorted to an approach based on a HMD, as shown in Figure 3.15, which offers an independent screen for each eye to achieve the stereoscopic effect. But, as it totally occludes the physical reality, including the users' body, additional feedback strategies are required, namely for users to know where their hands and arms are in relation to the virtual objects.

3.2.2. Implemented Techniques

In this study, we also wanted to determine which technique is more suited to manipulate virtual objects for each of the two types of virtual environments. To that, we chose the two techniques that attained the most promising results in our first study, the 6-DOF Hand and the Handle-Bar. Both allow users to perform transformations in 7 DOF: 3 for translations, 3 for rotations and 1 for uniform scaling. Still, these techniques were initially conceived for different scenarios, having different input requirements, and each show distinct advantages and disadvantages.

3.2.2.1. 6-DOF Hand

6-DOF Hand is essentially an extension to the Simple Virtual Hand [73] to also uniformly manipulate objects' scale. Simulating physical interactions, it uses both position and orientation tracking information of one of the users' hands to move and rotate a grabbed object, which will follow the hand with an exact mapping, as illustrated in Figure 3.16. Another grab gesture with the other hand, not necessarily intersecting the object, will enable the scaling transformation, through a rubber band metaphor [136]: increasing the distance between both hands will increase the object's size, and decreasing this distance will reduce it. While only grabbing the object with one hand, its position and orientation can be modified in 6 DOF at the same time. With both hands in a grab state, it is also possible to change its scale, achieving simultaneous 7 DOF manipulations.

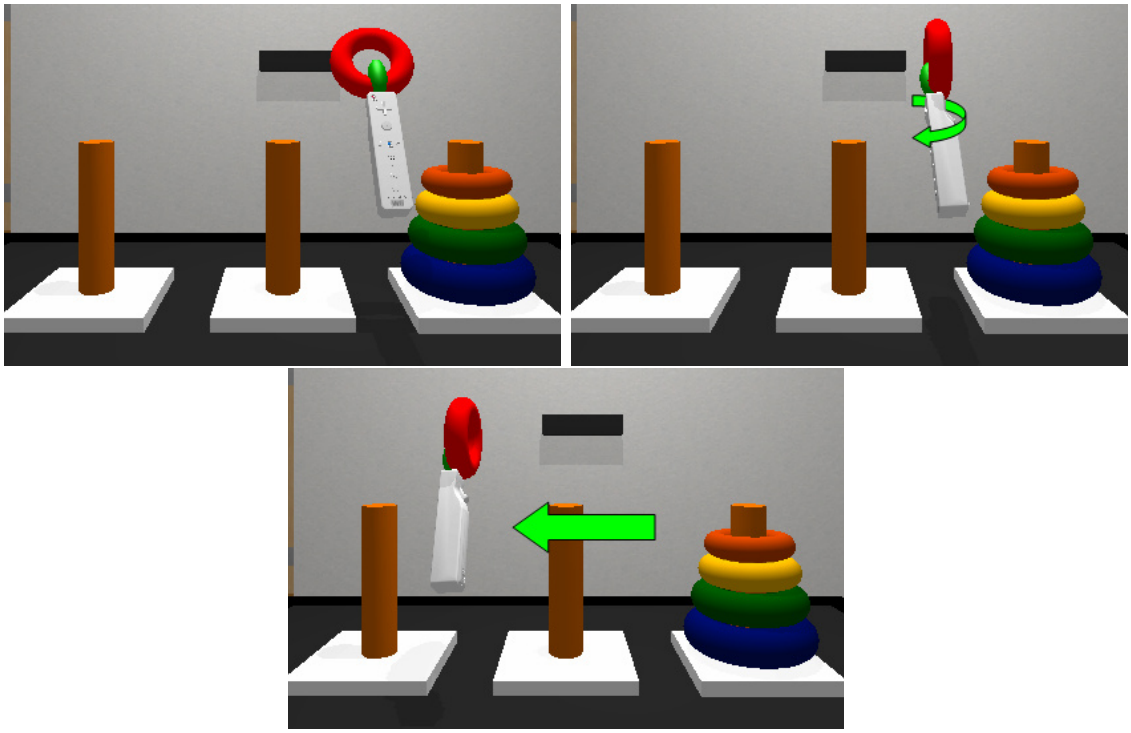


Figure 3.16: Using the 6-DOF Hand technique to manipulate a torus in our virtual environment. Hands are represented by the input devices they are holding.

3.2.2.2. Handle-Bar

The second technique implemented for our study is the Handle-Bar [116]. As opposed to the 6-DOF Hand, it only requires positional tracking. Performing a grab gesture with both hands will select an object intersected by the middle point between

both hands. Then interactions with a physical handle-bar are replicated: moving both hands in the same direction will move the object in 3 DOF, and moving them in opposite directions will rotate the object around the hands' middle point, as depicted in Figure 3.17. Again, the distance between the hands is used to control the object's scale. Since all transformations require both hands to be engaged, all 7 DOF can be controlled at the same time after starting a manipulation.

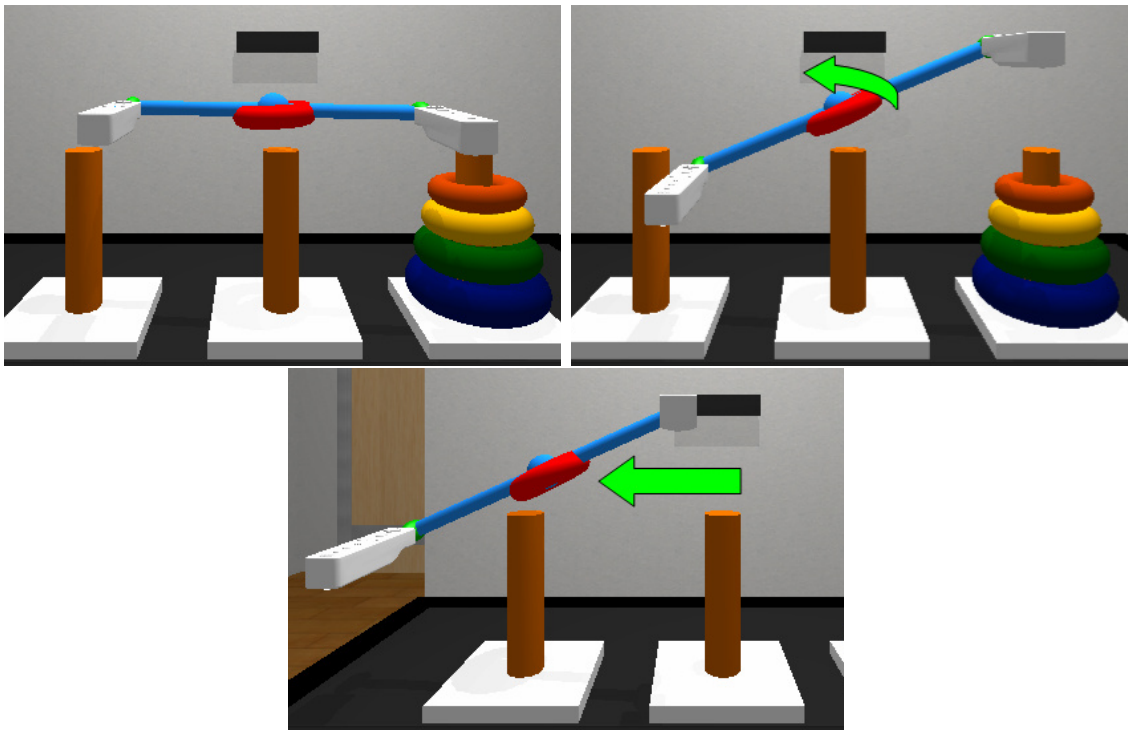


Figure 3.17: The Handle-Bar technique being used to manipulate an object in our virtual environment. Hands are represented by the input devices they are holding.

3.2.3. User Evaluation

In order to compare the implemented techniques and both virtual environments, we conducted a user evaluation. Each participant used the two techniques to manipulate virtual objects in each of the environments, thus experimenting a total of four conditions. To evaluate each condition, participants completed a set of tasks with different requirements, and both objective and subjective data was collected.

3.2.3.1. Procedure

All evaluation sessions were structured in a similar fashion. They were performed in our laboratory, a controlled closed environment without external influences. In the beginning of each test, participants were briefly introduced to all techniques and environments they would experiment. The order of the virtual environments was randomized, as well as the sequence of the techniques. However, participants always experimented the two conditions of the same environment in a row, and the order of the techniques carried on from the first environment to the second.

For each condition, it was shown how to translate, rotate and scale objects, and participants had a training period of five minutes to practice. The training scenario was a replica of the Tower of Hanoi game, but besides freely interacting, we recommended participants to take the disks from the initial stack and place them as portrayed in Figure 3.18, so that they performed transformations according to every DOF possible.

Then, participants were asked to execute a set of tasks. These tasks always followed the same order and required an increasing number of different transformations. Participants performed each task once with no time limit. After completing the tasks for each condition, participants were presented with a questionnaire to evaluate different aspects, such as easiness to perform each transformation and fun factor. Lastly, after finishing all tasks in every condition, participants filled out a profiling questionnaire. Each session had a duration of about 50 minutes on average.

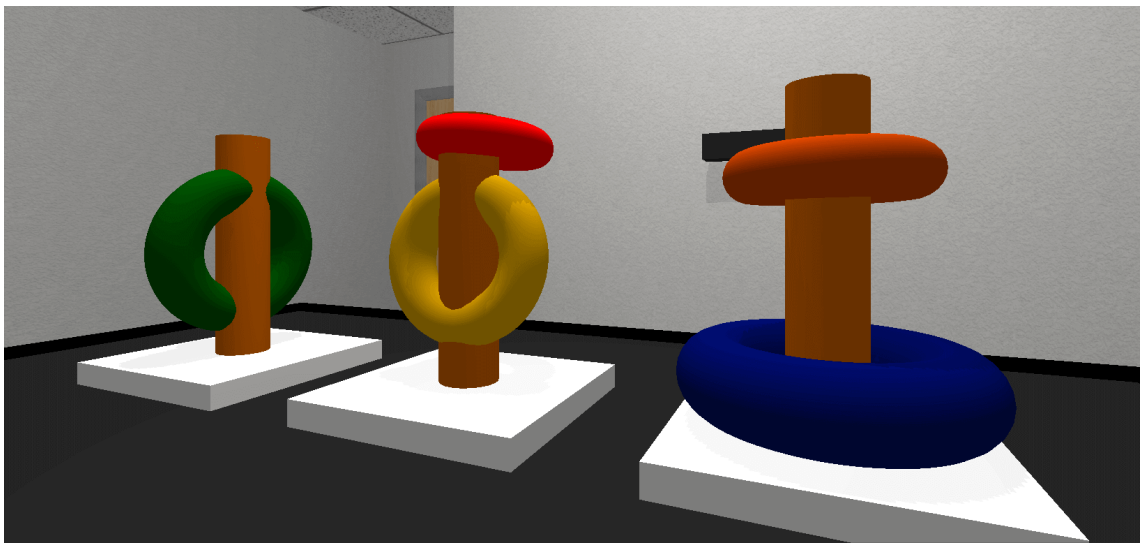


Figure 3.18: Objective for the training period.

3.2.3.2. Tasks

Similarly to the previous evaluation, we designed three docking tasks for users to complete. Here, we used a toy problem that consisted in connecting a power plug in the corresponding socket, as shown in Figure 3.19. Participants were instructed to complete the task with the highest accuracy possible. The starting position of the plug was randomly calculated, assuring that it was always visible and between the user and the socket, and at a fixed distance to the target, distributed along the three coordinates.

Tasks were executed following an incremental difficulty. On the first task, participants only needed to translate the plug. On the second, they had to both translate and rotate it. The third task required participants to perform translation, rotation and scaling transformations on the object. Although not all transformations were needed to achieve the target placement on every task, they were always active and no restrictions were applied.

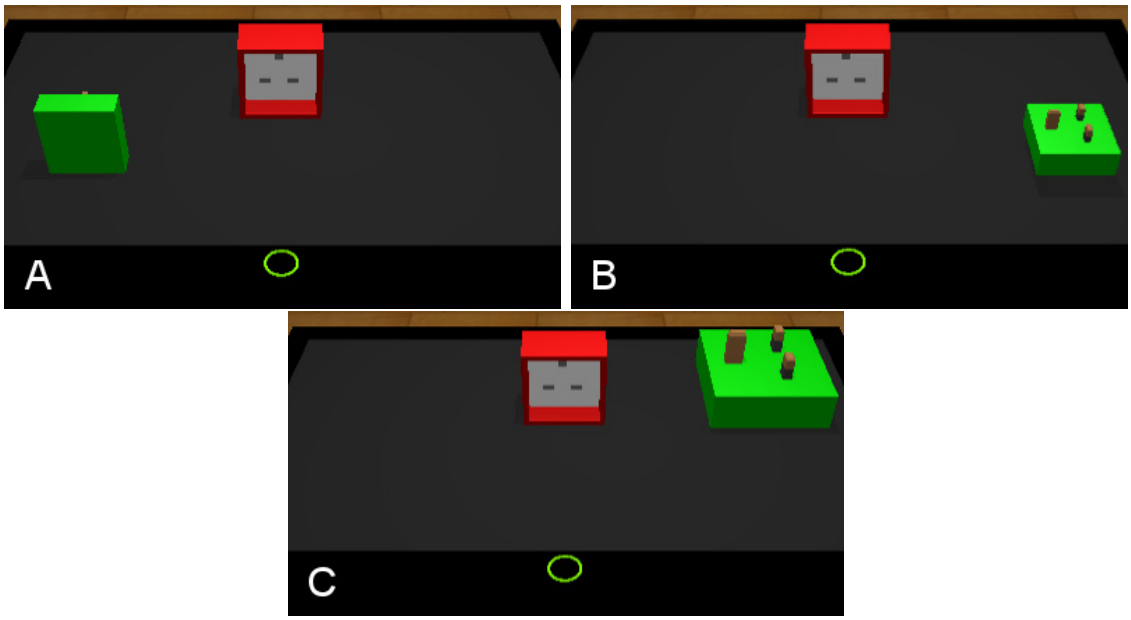


Figure 3.19: Tasks for the second user evaluation. All consisted in placing the green power plug into the red socket. Task 1 (A) needs only translation; task 2 (B) requires translation and rotation; task 3 (C) involves all three transformations.

3.2.3.3. Setups and Prototypes

We developed two prototypes in order to compare the two object manipulation techniques and the two virtual environments. They shared some characteristics and differed in others.

Hardware Setups

All the hardware used in our prototypes is depicted in Figure 3.20. Both IVE and SIVE configurations used the same input for the spatial interactions, and differed mostly in the visualization devices. The visualization in SIVE was achieved in a similar fashion to the setup of the previous study, using a stereoscopic tabletop. This time, however, instead of a stereoscopic projector, it used a 3DTV (model: Samsung UE55F8000), which offered an improved image. The IVE setup was built with a HMD, the Oculus Rift DK2.

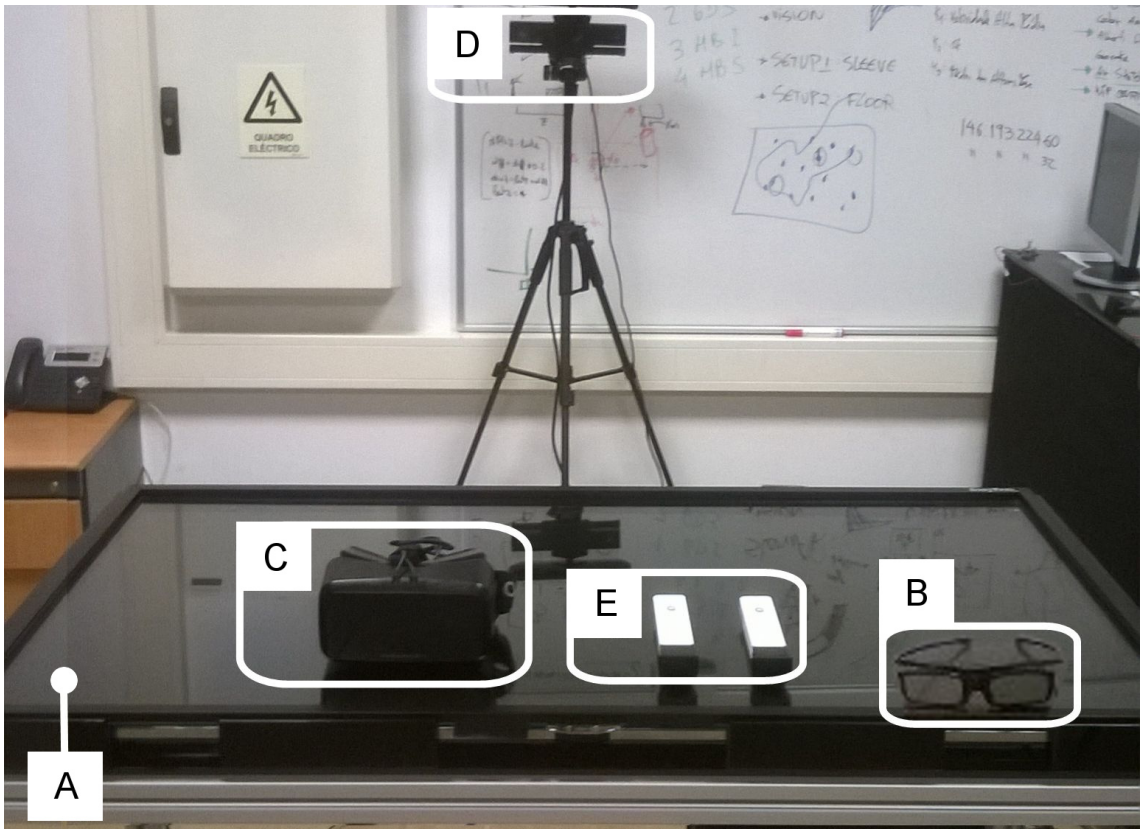


Figure 3.20: Hardware used in our prototypes: Samsung 3DTV (A) and corresponding active-shutter glasses (B); Oculus Rift DK2 (C); Microsoft Kinect v2 (D); and Wiimote controllers with Motion Plus (E).

For user tracking, both setups used a Microsoft Kinect v2 depth camera to gather head, hands and limbs' position. In the IVE, the head's position was combined with the orientation given by the headset to generate users' perspective. Hands' orientation was tracked with hand-held devices, two Wiimote controllers with the Motion Plus adapter. The grab gesture was mapped to the trigger on the controllers. This choice was made because, although finger-based interactions can indeed be more natural and realistic, controller-based interactions are faster and more robust [92, 127]. In both setups, the interaction with the virtual objects was only made in the space above the tabletop.

Virtual Environment

The virtual environment used in our prototypes was developed used the Unity3D engine, and recreated our laboratory, so that the perceived surroundings were as similar as possible in both configurations. Tasks' objects, the plug and the socket, were placed above the tabletop. In the SIVE, these were the only objects displayed on the screen, along with two ellipsoid cursors that appeared to be placed on top of the physical controllers, as suggested by Lubos et al [76]. In the IVE, since users could not see the physical world, we used virtual models of the actual controllers

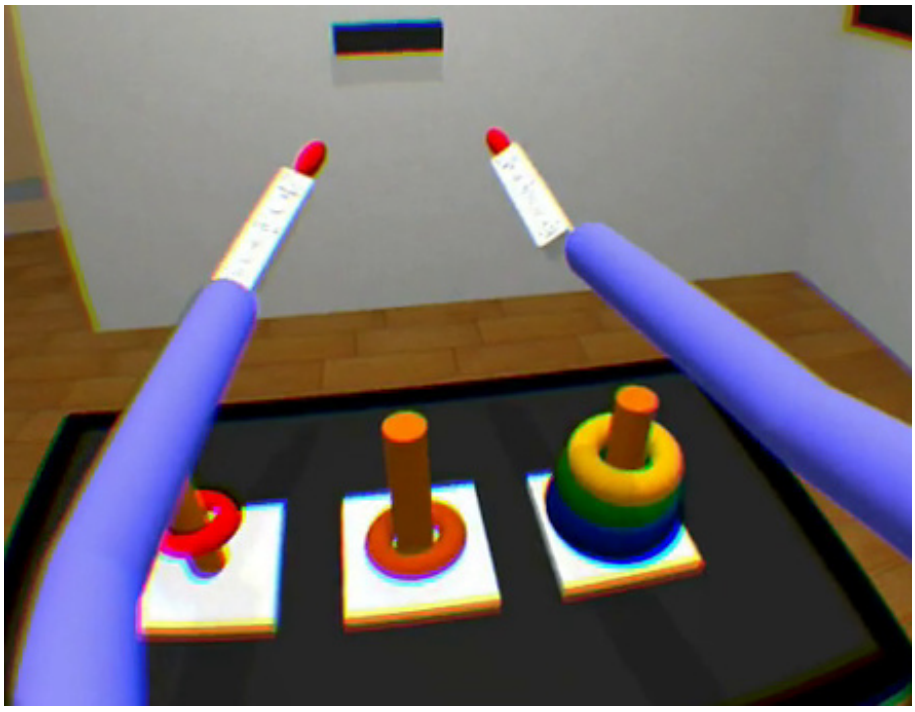


Figure 3.21: Screen capture of our immersive virtual environment.

paired with the same ellipsoid cursors, and cylinders representing users' arms, as shown in Figure 3.21.

For the 6-DOF Hand technique, objects could be selected while intersected by a cursor and pressing the corresponding trigger button. In the Handle-Bar, object selection was made with the middle point of the two hands, represented by a small blue sphere. After selecting an object, a virtual blue bar was also shown, connecting both hands and going through the object. No gravity and no collisions between objects were enabled, and the only virtual object users could grab and manipulate was the power plug.

3.2.3.4. Participants

The evaluation counted with the participation of 20 people, one female and 19 males, whose ages were between 19 and 35 years old. Only four participants hadn't had previous experience with stereoscopic installations, and eight had never interacted in virtual reality. 19 participants had already used spatial input devices for entertainment: ten had played with the Microsoft Kinect, 14 with the Wiimote and eight with the Playstation Move.

3.2.4. Results and Discussion

During tasks' execution, we logged task performance measures. After finishing them, we asked participants to complete a set of questionnaires in order to collect user preferences. We compared the registered measures according to two factors: immersiveness of the system and the manipulation technique used.

3.2.4.1. Task Performance

To analyze task performance, we logged completion time and placement precision. For precision, we considered both position and orientation errors in relation to the target. To find statistical significant differences, we used the two-way repeated measures ANOVA.

Completion Time

Tasks' completion times for the four conditions are depicted in the chart of Figure 3.22. For the first task, we found that the immersiveness of the system significantly impacted completion time ($F(1, 12) = 26.752$, $p < 0.0005$), with the IVE (average: 26.3s) being faster than the SIVE (average: 40.4s), most likely due to the improved visualization of the virtual content. The technique used did not have any effect, which is possibly related to the easiness of the task, and we did not find any interaction between the system and the technique.

On the other tasks, however, we only found statistically significant differences between the techniques (Task 2: $F(1, 14) = 74.217$, $p < 0.0005$; Task 3: $F(1, 10) = 23.417$, $p = 0.001$), where the 6-DOF Hand (Task 2 average: 33.2s; Task 3 average: 38.8s) showed better results than the Handle-Bar (Task 2 average: 78.7s; Task 3 average: 61.4s). The higher resemblance of 6-DOF Hand to everyday physical manipulations may have contributed for these results, as well as the need to perform additional transformations to rotate the object around the axis of the Handle-Bar.

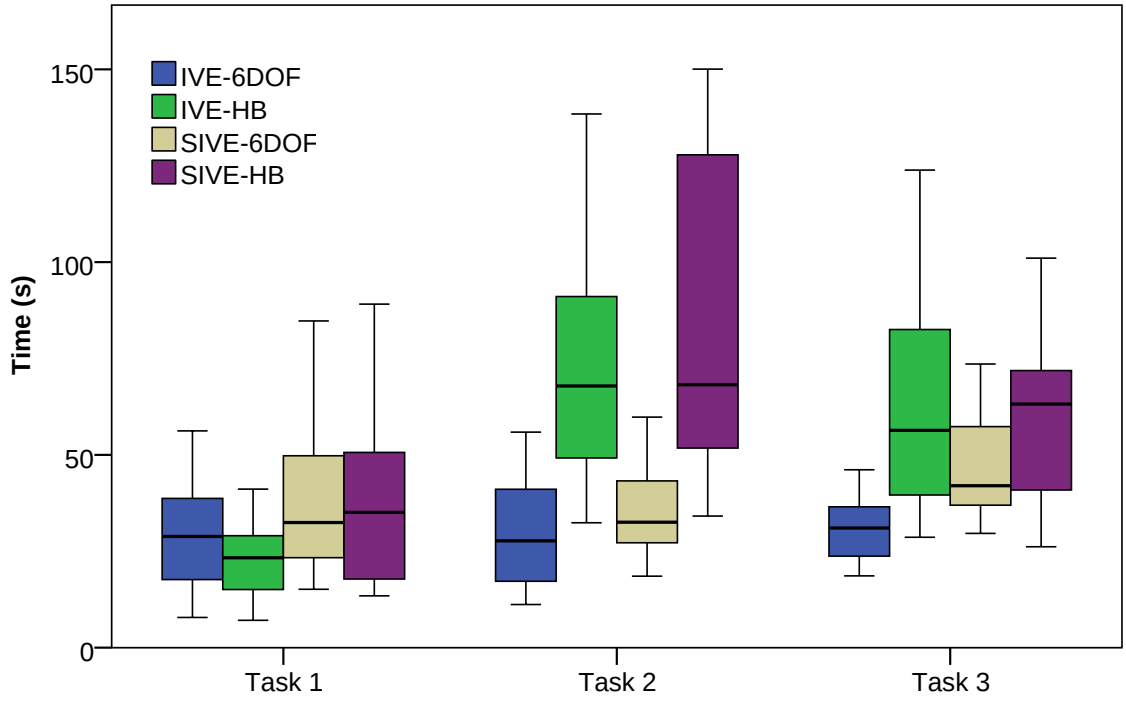


Figure 3.22: Time to complete the three tasks using the four conditions. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

Position Error

The position error attained with each condition for the three tasks is shown in the chart of Figure 3.23. In the first task, we found that the configuration used had a significant effect ($F(1, 10) = 17.481$, $p = 0.002$). The IVE (average: 22mm) allowed more precise placements than the SIVE (average: 40mm). However, in the second task, the SIVE (average: 32mm) attained a smaller error ($F(1, 10) = 5.752$, $p = 0.037$) than the IVE (average: 49mm). These contradictory results suggest that there is no clear winner for reducing position error.

Still for the second task, we also found that the technique used affected the position error ($F(1, 10) = 9.156$, $p = 0.013$), with the 6-DOF Hand (average: 29mm) being more accurate than the Handle-Bar (average: 52mm). This might be once again related to participants being more acquainted with this type of manipulation, even though the Handle-Bar can prevent objects being occluded by the users' hands. For the third task, no statistically significant results were found.

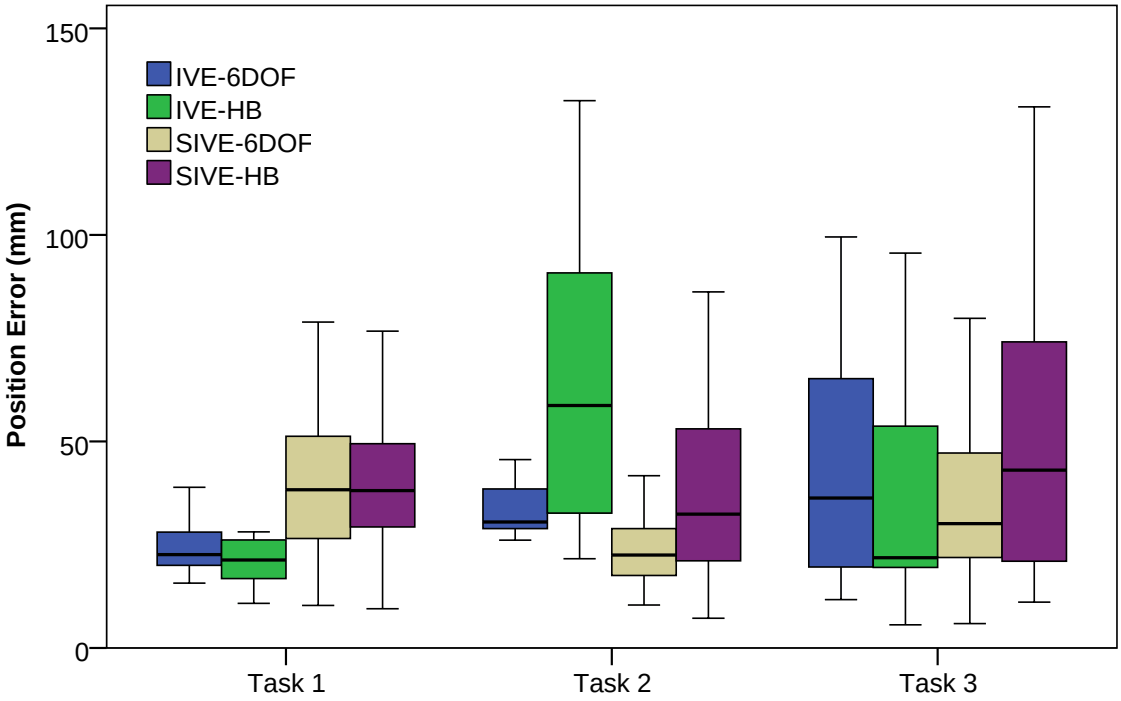


Figure 3.23: Position error attained in the three tasks using the four conditions. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

Rotation Error

The chart of Figure 3.24 reveals the rotation error of all conditions in each task. The only statistically significant result found was that the technique used had an effect on the rotation error achieved for the first task ($F(1, 11) = 12.570$, $p = 0.005$). Here, the Handle-Bar (average: 2.3°) reduced object's orientation difference to the target in relation to the 6-DOF Hand (average: 3.3°). This suggests that the Handle-Bar might be better in preventing unwanted rotations than the 6-DOF Hand when only translations are required.

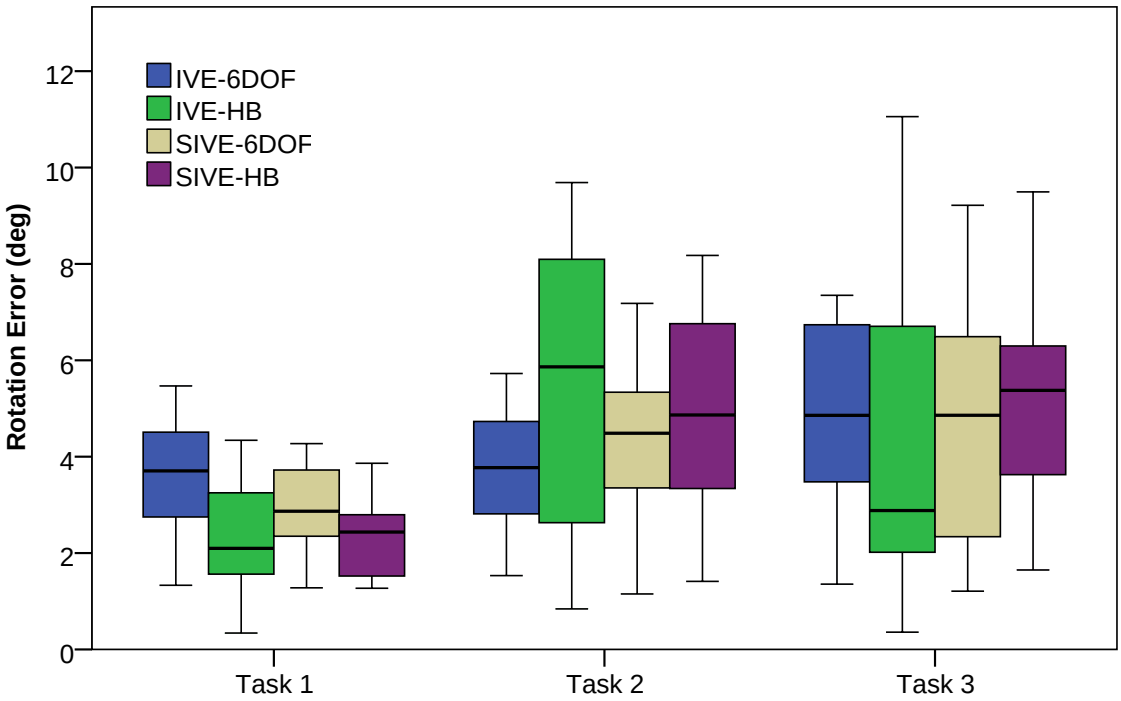


Figure 3.24: Rotation error attained in the three tasks using the four conditions. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

3.2.4.2. User Preferences

We gathered participants' opinions regarding the use of the techniques in both environments through questionnaires. These consisted of several questions that were answered using Likert scales with five points, being five the favorable value. Participants' answers are shown in Table 3.2. We resorted to the two-way repeated measures ANOVA to assess the significance of the results. When a statistically significant interaction between the technique and the environment existed, we used the

Technique	6-DOF Hand	Handle-Bar	6-DOF Hand	Handle-Bar
Environment	IVE	IVE	SIVE	SIVE
Overall *	4,5 (1)	4 (1)	4 (2)	3 (1)
Translation *	5 (1)	4 (1)	4 (0)	4 (2)
Rotation *	4 (1)	3 (1)	3 (1)	2,5 (3)
Scaling *	5 (2)	4 (1)	4 (2)	3,5 (2)
Fun *	5 (1)	4 (2)	3 (2)	3 (2)

Table 3.2: Participants preference for each pair technique-environment regarding different criteria (median, interquartile range). * indicates statistical significance.

Wilcoxon Signed-Rank test with the Bonferroni correction (presented p-values are corrected).

For the overall ease of use, we found an interaction between the techniques used and environments experimented ($F(1, 19) = 6.577$, $p = 0.019$). Participants strongly agreed that using the 6-DOF Hand in the IVE was easier than in the SIVE ($Z = -3.066$, $p = 0.008$), and that the 6-DOF Hand offered better manipulations than the Handle-Bar in the IVE ($Z = -3.220$, $p = 0.004$).

Going into detail, the IVE was preferred over the SIVE to perform object translation ($F(1, 19) = 9.816$, $p = 0.005$) and scaling ($F(1, 19) = 7.912$ $p = 0.011$). The better visualization provided by the HMD may have aided in these transformations. For rotations, there was an interaction between techniques and environments ($F(1, 19) = 9.320$, $p = 0.007$), and participants agreed once again that the 6-DOF Hand was easier to use in the IVE than in the SIVE ($Z = -3.542$, $p < 0.0005$). Also, the Handle-Bar was harder to use than the 6-DOF Hand in the IVE ($Z = -3.779$, $p < 0.0005$). This might have been caused by the need to perform additional transformations to rotate the object about the axis of the Handle-Bar.

Finally, concerning the fun factor, an interaction between techniques and environments existed ($F(1, 19) = 4.524$, $p = 0.047$). Participants considered that both the 6-DOF Hand ($Z = -3.800$, $p < 0.0005$) and the Handle-Bar ($Z = -3.080$, $p = 0.008$) were more amusing to use in the IVE than in the SIVE. Moreover, within the IVE, the 6-DOF Hand was better than the Handle-Bar ($Z = -3.082$, $p = 0.008$). For the SIVE, no significant differences were found between the 6-DOF Hand and the Handle-Bar.

3.2.5. Lessons Learned

Motivated by recent developments in visualization devices for immersive virtual environments, we set out to compare how such type of environment fared against the semi-immersive environment of our previous study, for object manipulation purposes. For this, we created two prototypes that co-locate users' hands and the virtual objects, one using a HMD and the other a stereoscopic tabletop. Moreover, since the better two manipulation techniques of that study, the 6-DOF Hand and the Handle-Bar, were developed for different settings, we included both in this evaluation in order to assess how they perform in the two environments. As such, participants of our user evaluation experimented manipulations with four different conditions.

Overall, the 6-DOF Hand performed better than the Handle-Bar, mostly due to how they perform rotations, but the Handle-Bar helped prevent rotation error on purely translation tasks. Participants' opinions revealed that, when in immersive environments, the 6-DOF Hand was easier to use than the Handle-Bar, because it is more natural. Participants also agreed that performing manipulations in the IVE was preferred over the SIVE. That might be related to the content's improved perception in the IVE, since users can move more freely around the objects and see them from different points of view. In the SIVE, viewing angles are more limited.

These results, besides corroborating the benefits of higher levels of immersion, as stated by Bowman and McMahan [19], also show the need for more precise manipulation techniques. The average position and orientation errors attained by participants were 37 mm and 4.0 degrees, respectively, which is not enough to compete against traditional desktop interactions.

3.3. Chapter Summary

In this chapter, we presented two user evaluations of mid-air object manipulation techniques, involving translation, rotation and uniform scaling. In the first, we compared five manipulation techniques for virtual objects placed above a stereoscopic tabletop, in a semi-immersive virtual environment. These techniques were based on existing research, four were mid-air and one resorted to multi-touch. In this study, the 6-DOF Hand, an approach based on the Simple Virtual Hand [73], and

the Handle-Bar, which only requires positional hand-tracking, positively stood out from the others.

Our second study focused on assessing the differences in manipulating virtual objects between fully-immersive and semi-immersive virtual environments. For that purpose, we compared the two best techniques from the previous study in two different setups. One used a head-mounted display and the other was comprised of a stereoscopic tabletop. The 6-DOF Hand technique achieved a better performance overall, and the fully-immersive setup provided an improved visualization, which made tasks easier and was considered more pleasant to use.

The main finding of the studies presented in this chapter is that, if there are no restrictions, the best existing combination for virtual object manipulation in mid-air is an approach similar to the 6-DOF Hand in a fully-immersive environment. This defined the direction followed in the works of the next chapters towards fully-immersive virtual environments, as well as the baseline technique used for evaluating methods to improve precision in mid-air manipulations.

4

Precise Object Manipulation

Manipulation of virtual objects in mid-air using spatial input allow for highly natural metaphors, letting users grab, move and rotate objects in a similar way to how it is done in the physical world. Nonetheless, mid-air gestures compromise object placement accuracy, whether due to limitations in tracking solutions or to human dexterity itself.

In this chapter, we study ways of improving accuracy in mid-air virtual object manipulations. It is our belief that DOF separation can be beneficial in mid-air, after the positive results it showed in mouse and touch-based interfaces, as well as a proper utilization of scaled down users' movements. To assess this, we devised a set of user studies, which we describe. Following evaluations' results of the previous chapter, these studies are focused on spatial manipulations within IVEs.

In Section 4.1, we present an evaluation comparing three manipulation techniques: one follows a direct 6-DOF approach, the second scales users' movement, and the third uses mid-air virtual handles for DOF separation. Taking into consideration the attained results, we propose WISDOM, a novel mid-air manipulation technique, in Section 4.2. To develop it, we gathered the positive aspects of each one of the three

techniques evaluated. We then report results of a second user evaluation, where we validate WISDOM. Lastly, in Section 4.3, we explore custom transformation axes, as an alternative to the traditional fixed frames for single DOF control. For this, we propose a second manipulation technique, MAiOR, which we compared against a direct approach and single DOF widgets.

4.1. Exploring DOF Separation

For both mouse and touch interfaces, it has been shown that separating DOF leads to better performance when compared to direct approaches, but this was motivated by needed mapping between the 2D input and 3D output, which does not occur in mid-air. However, since it may prevent unintended and unexpected transformations, full and explicit DOF separation may also be useful for mid-air object manipulation within IVEs.

To reduce the simultaneous DOF being controlled, most mouse based interfaces rely on widgets for object manipulation [60, 32, 111]. For touch enabled surfaces, albeit allowing users to directly interact with the objects displayed, researchers found out that DOF separation led to better performance on object manipulation tasks, having turned once again to virtual widgets to clearly and undoubtedly select transformations and axes [30, 16, 135].

To improve users' accuracy in mid-air interactions, researchers already tried to either scale down hand motions [43] or move the viewpoint closer to the object being manipulated [99], but without regard to DOF separation. On the other hand, approaches based on virtual widgets have already been proposed [96, 97]. However, these techniques do not have promising results, possibly because widgets used in these approaches are very different from those used in mouse and touch based interfaces, being more complex, not allowing controlling a single DOF at a time and not using common reference frames, such as object or world axes. We believe that clear DOF separation in mid-air scenarios, using familiar virtual widgets, might improve users' performance in object manipulation tasks.

In this section, we present an evaluation comparing a mid-air implementation of virtual handles for single DOF manipulation against approaches following exact and scaled mappings. To restrict the scope of this evaluation, we will focus on positioning tasks requiring only translation and rotation. We start by presenting the implemented manipulation techniques. We follow with a description of the

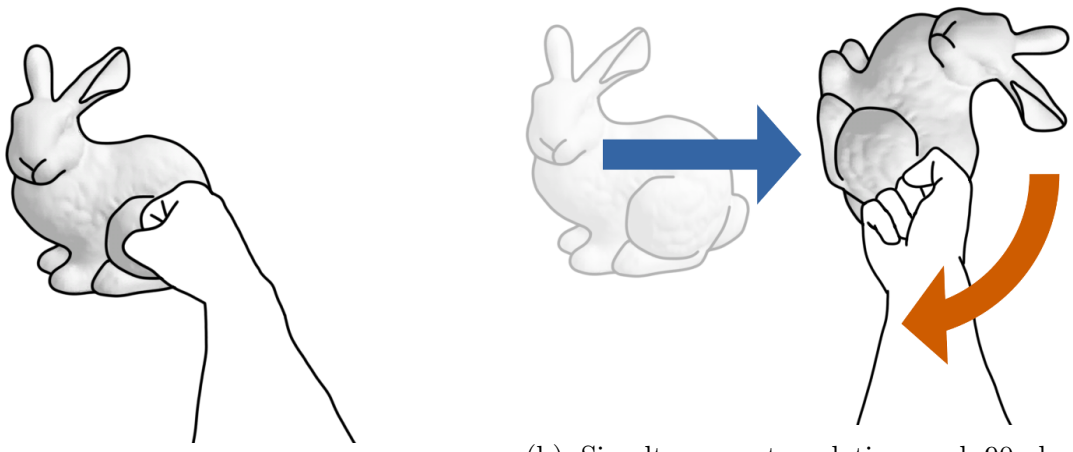
user evaluation, namely the method and tasks, the prototype developed, and the participants that took part in it. Then, we present and discuss the obtained results and the lessons learned.

4.1.1. Techniques Implemented

For our evaluation, we implemented three techniques based on the literature. The first is a direct approach in which all transformations performed by the user hand are directly applied to the object, the second follows scaled transformations based on the user movement's speed, and the third consists of spatial widgets for separating DOF. All implemented techniques provide 6 DOF transformations: three for translation and three for rotation.

4.1.1.1. Simple Virtual Hand

The Simple Virtual Hand [73] (SVH) mimics interactions with physical objects as closely as possible. This direct approach follows an exact mapping and uses all 6 DOF information from users' hands. It is often used as a baseline for evaluations of other techniques [43, 97]. This technique consists of grabbing an object directly, moving it to a new location and/or rotating it, and then releasing. After being grabbed, the object directly follows the movement of the hand: dragging changes object's position and wrist's rotation controls object's rotation. All transformations



(a) Initial State.

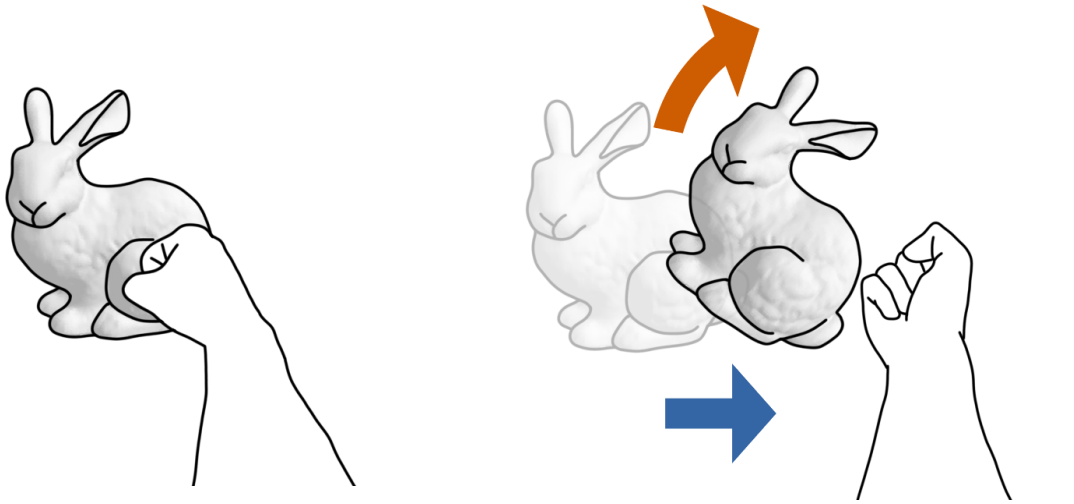
(b) Simultaneous translation and 90 degrees rotation.

Figure 4.1: Manipulation of an object using an exact mapping with Simple Virtual Hand approach.

are simultaneously applied to the object, as pictured in Figure 4.1. The grabbed point in the object will remain the center of all transformations during the entire manipulation, until the object is released.

4.1.1.2. PRISM

We implemented the PRISM technique as presented by Frees et al. [42]. This technique aims in improving accuracy of direct manipulation, switching between a precise and a direct mode according to the current velocity of users' hands. Hand's movement in each coordinate axis is scaled down when users move their hands slower than a pre-defined threshold in that axis. We used the threshold value proposed by the original authors. This scaling results in an offset between the hand and the object being manipulated, that can be canceled by moving hands faster than the same threshold. We also included rotations later proposed by same authors [43], which follows the same premise from translations, scaling down slow wrist rotations. Similarly to 6DOF technique, both translations and rotations can be performed simultaneously, as exemplified in Figure 4.2.



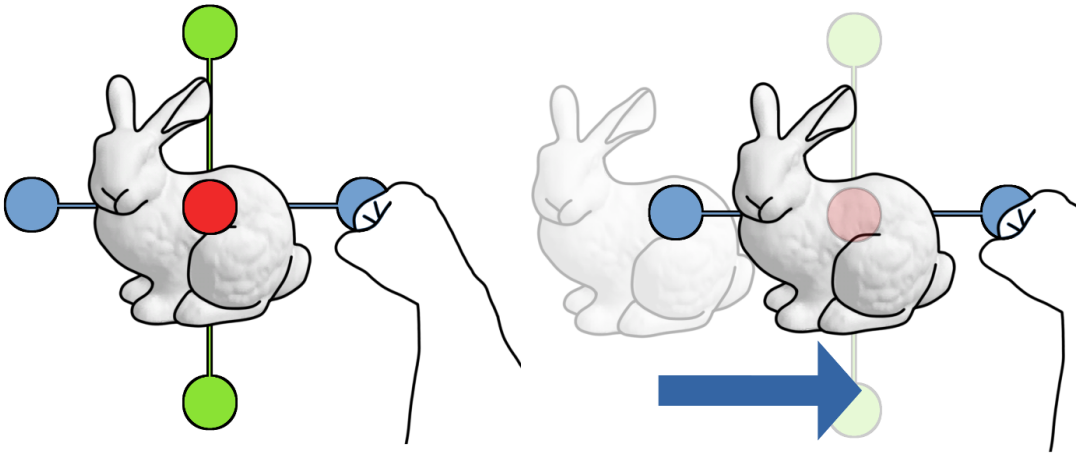
(a) Initial State.

(b) Simultaneous scaled translation and rotation, with accumulated offset.

Figure 4.2: PRISM technique.

4.1.1.3. Widgets for DOF Separation

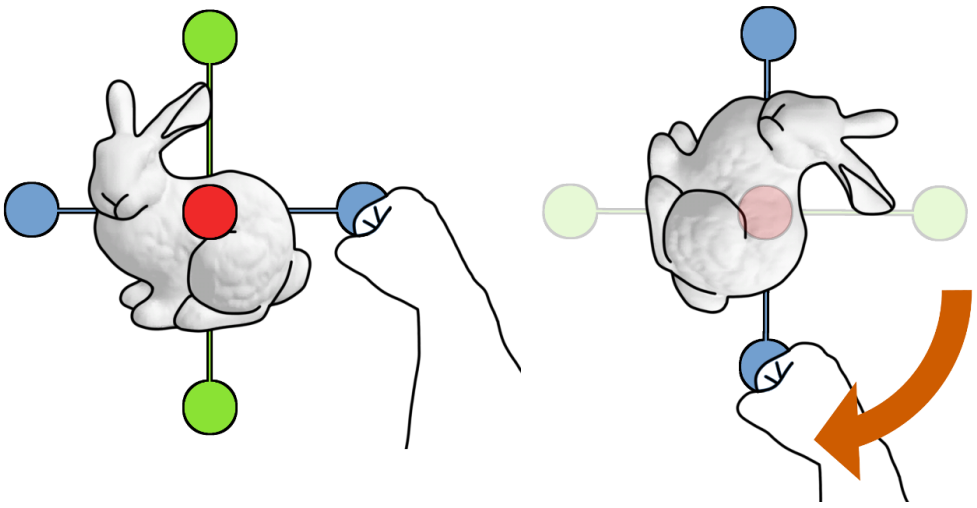
Widget based manipulations are widely used in mouse and keyboard 3D user interfaces. Our implementation, as opposed to the previous two, strictly follows DOF



(a) Initial State. (b) Translation.

Figure 4.3: Translation in Widgets technique.

separation. Not only translation and rotation operations are treated independently, users can only manipulate 1 DOF at a time. We used a representation similar to that introduced by Conner et al. [32]. Users can grab the sphere connected to the desired axis and move the hand along the axis to trigger object translation (Figure 4.3). For rotations, the approach is similar, but the hand movement is performed around the target axis (Figure 4.4). The decision to either perform a translation or rotation, is made based on the hand's path after 10 cm. Selected transformation and axis remain locked until a release gesture.



(a) Initial State. (b) Rotation.

Figure 4.4: Rotation in Widgets technique.

4.1.2. User Evaluation

In order to assess if DOF separation also benefits spatial interactions for 3D object manipulation in IVEs, we conducted a user evaluation comparing the techniques described above.

4.1.2.1. Procedure

All user sessions followed the same structure, each lasting approximately 45 minutes. The experiment was performed in our laboratory, with a controlled environment. We started by introducing the experiment the participant was about to perform, followed by a brief description of the techniques being evaluated. The techniques were performed in alternated order, following a Latin square design, assuring that each one was experienced in every possible permutation, in order to avoid biased results. For each technique we played a video showing how to apply transformations to the object with it. After the video, participants had a training period of three minutes, or less if they considered themselves to be already acquainted, to explore the approach in a dedicated environment, showed in Figure 4.5. Following the practice period,

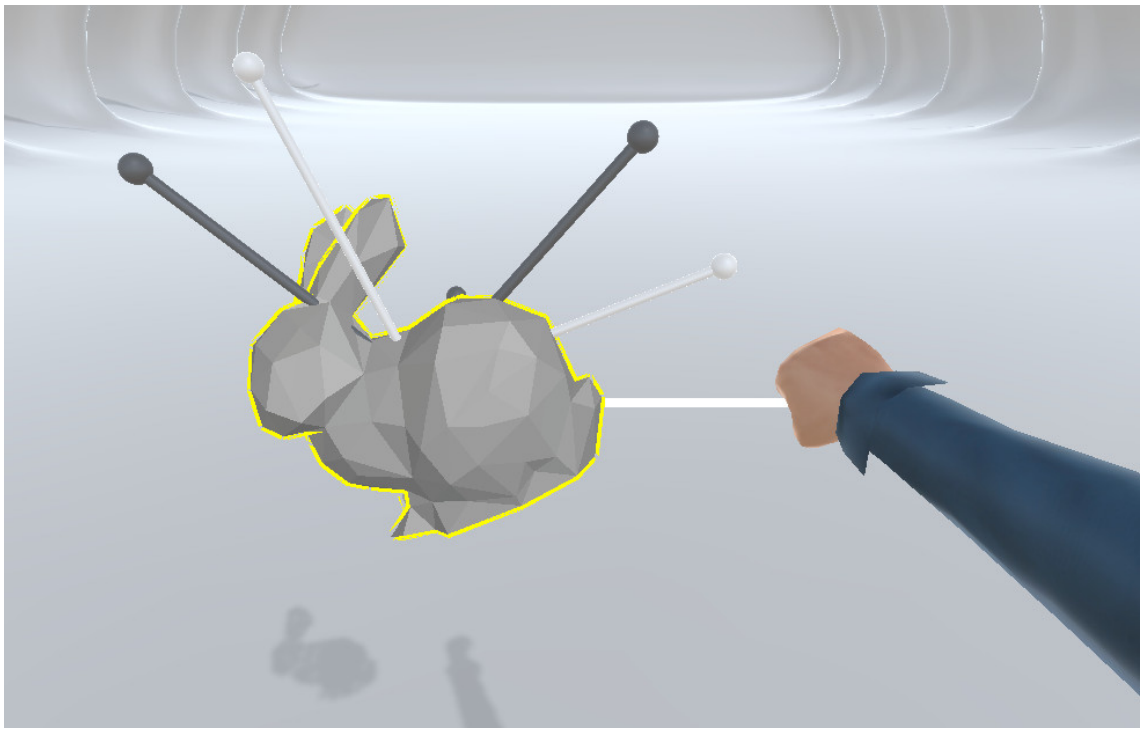


Figure 4.5: Interacting with an object in our virtual environment during the training period, with the PRISM technique.

we asked participants to perform six tasks, described next. After completing each technique's tasks, participants fulfilled a questionnaire regarding distinct aspects of the interaction. The experiment concluded with a profiling questionnaire.

4.1.2.2. Tasks

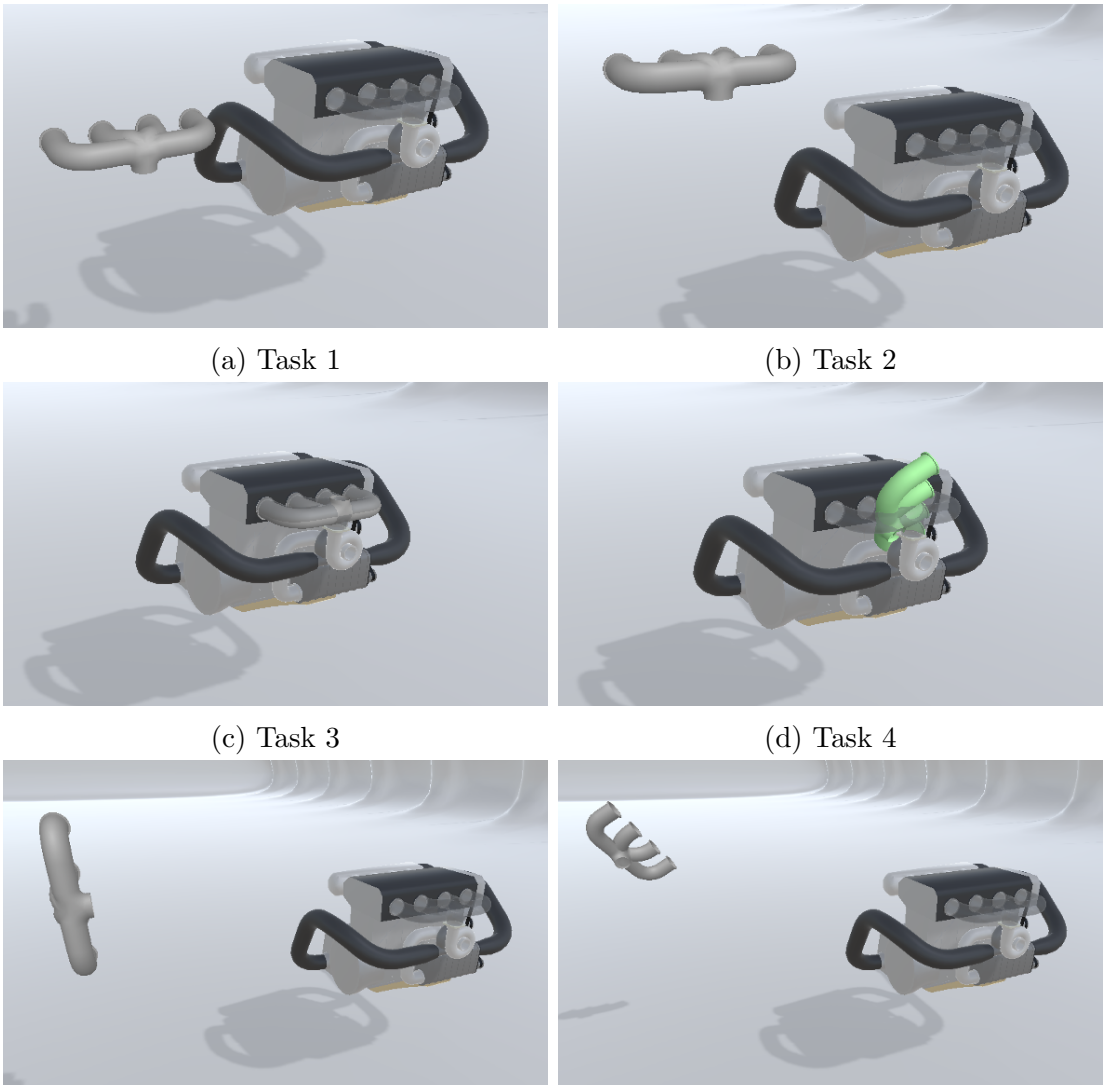
As we mentioned in the previous section, we requested participants to complete a set of six tasks for each technique. All consisted in a docking task, where participants had to put the exhaust pipes in the right place of a car engine[1]. The engine's model had a semi-transparent replica of the pipes showing the only possible target position and orientation, as depicted in Figure 4.6. To prevent excessively long sessions, each task was limited to a maximum of three minutes. After reaching time limit we informed participants they could stop, and we considered the attained position and orientation as final.

For the first task (Figure 4.6a), the object to be manipulated begun with the correct position along both YY and ZZ scene axes and orientation, only with an incorrect position according to the X coordinate. Similarly to the previous task, the second task (Figure 4.6b) started with the object with the correct orientation, however its position was incorrect along all three coordinates. The third task (Figure 4.6c) consisted in only rotating the object around the Z axis, while the fourth task (Figure 4.6d) implied rotation around an arbitrary axis, requiring no translation as well. The fifth task (Figure 4.6e) required the object to be rotated around the Z axis and translated along both XX and YY axes. Finally, in the last task (Figure 4.6f), participants had to apply full 6-DOF transformations to the object. Although some tasks required only one kind of transformation (translation or rotation), none was restricted, as we did not intend to modify any technique in order to accommodate a specific task.

4.1.2.3. Setup and Prototype

We developed a prototype in which participants could experiment with the implemented techniques during the evaluation.

[1]Original 3D model uploaded to SketchUp's 3D Warehouse by the user M-Speed. Url: `http://3dwarehouse.sketchup.com/model/u2a59c8f9-277f-45cb-8d5a-daedd9dcfd87/IPT-SB4-20L-Twincharged-4-cylinder`, accessed 17-February-2018.



(a) Task 1 (b) Task 2

(c) Task 3 (d) Task 4

(e) Task 5 (f) Task 6

Figure 4.6: Tasks performed by the participants.

Hardware Setup

Our setup comprises non-invasive and affordable full body user tracking with three depth cameras Microsoft Kinect v2, shown in Figure 4.7. One of them was placed facing the user while the remaining ones lied on each side, 90 degrees from the first one. To combine all three cameras' data into a single user skeleton, we use our own tracking solution, the Creepy Tracker, described in Appendix A.

Since Microsoft Kinect fails in providing reliable hand orientation data, and we wanted to provide an almost hands-free experience, we developed an wireless custom made device to better acquire such data, depicted in Figure 4.8. It uses an IMUduino, an Arduino based circuit board, that incorporates an inertial measure-

Figure 4.7: Panorama of our laboratory, showing the 3 Microsoft Kinect v2 used for positional tracking in our setup.

ment unit (IMU) and a Bluetooth LE modules. The IMU is composed of gyroscope, accelerometer and digital compass sensors for accurate 3 DOF orientation tracking. We attach the device to the user's dominant hand using an acrylic clip, which assures it does not fall when the hand is opened. A pressure pad detects if the hand is open or closed. For the visualization component, we used a Gear VR with a Samsung Galaxy S6, connected via Wi-Fi to our tracking server.

Virtual Environment

We developed our prototype using Unity3D engine, with gravity and objects' collision disabled. Users' had a full body avatar, shown in Figure 4.8, whose limbs and hands were animated using data from the Kinects and the IMUduino device, re-

Figure 4.8: Our custom made device for tracking hand's rotation and its open / grab state (left) and users' avatar in out virtual environment (right).

spectively. The exhaust component of the engine was the only object in our virtual environment that could be grabbed and transformed. For improved user feedback while grasping, the object becomes transparent, revealing the penetrating portion of the hand, as suggested by previous research [105]. To guide participants during evaluation tasks, we make the object gradually turn green as it approaches the target position and orientation, as it can be seen in Figure 4.6d. For the PRISM technique, as suggested by the proposing authors [43], resulting offsets between the user's hands and the virtual object, due to scaled down motions, are represented by a white line for translations, and two sets of axis for rotations, as visible in Figure 4.5.

4.1.2.4. Participants

We counted with the participation of 21 people (five female), between the ages of 18 and 50 years old, with the great majority (62%) between 18 and 25. Most had at least a BSc degree (86%), while the remainder are finishing it. More than half (52%) had never experienced a VR setting, and 43% use some kind of gesture recognition systems more than once a month, such as XBox Kinect, Wii Remote or Playstation Move. Only 28% of participants use 3D modelling systems at least once a month.

4.1.3. Results and Discussion

During our experiment, we collected both task performance data, through logging mechanisms, and user preferences, by asking participants to fill out questionnaires. We used Shapiro-Wilk test to assess data normality. We then ran the repeated measures ANOVA test to find significant differences in normal distributed data, and Friedman non-parametric test with Wilcoxon-Signed Ranks post-hoc test otherwise. In both cases, post-hoc tests used Bonferroni correction (Presented sig. values are corrected).

4.1.3.1. Task Performance

We measured time taken by participants to fulfill each task, as well as object placement error. Time taken for all tasks, in seconds, is depicted in the graph of Fig-

ure 4.9. Regarding errors, we registered both position error, in millimeters (Figure 4.10), and rotation error, in degrees (Figure 4.11).

For the translation only tasks, we found statistically significant differences in completion time (Task 1: $\chi^2(2) = 25.368$, $p < 0.0005$; Task 2: $F(1.611, 30.604) = 9.025$, $p = 0.002$). For the first task, post-hoc test revealed Widgets approach (average: 25s) to be faster than both SVH (average: 59s, Z = -3.542, $p < 0.0005$) and PRISM (average: 90s, Z = -3.823, $p < 0.0005$), and SVH to be faster than PRISM (Z = -3.267, $p = 0.003$). In the second task, PRISM (average: 102s) was significantly slower than Widgets (average: 49s, $p = 0.008$) and SVH (average: 71s, $p = 0.028$). For position error, differences were also found (Task 1: $F(1.851, 24.066) = 17.474$, $p < 0.0005$; Task 2: $F(1.359, 14.946) = 6.653$, $p = 0.015$), with Widgets (Task 1 average: 3.3mm; Task 2 average: 5.2mm) outperforming SVH in the first task (average: 15.0mm, $p < 0.0005$) and PRISM on both first (average: 10.7mm, $p = 0.002$) and second tasks (average: 12.2mm, $p = 0.003$). The technique used also influenced rotation error (Task 1: $\chi^2(2) = 24.500$, $p < 0.0005$; Task 2: $\chi^2(2) = 15.000$, $p = 0.001$), with Widgets (Task 1 average: 0.0^o, Task 2 average: 0.0^o) achieving lower error than SVH (Task 1: average: 11.7^o, Z = -3.724, $p < 0.0005$; Task 2:

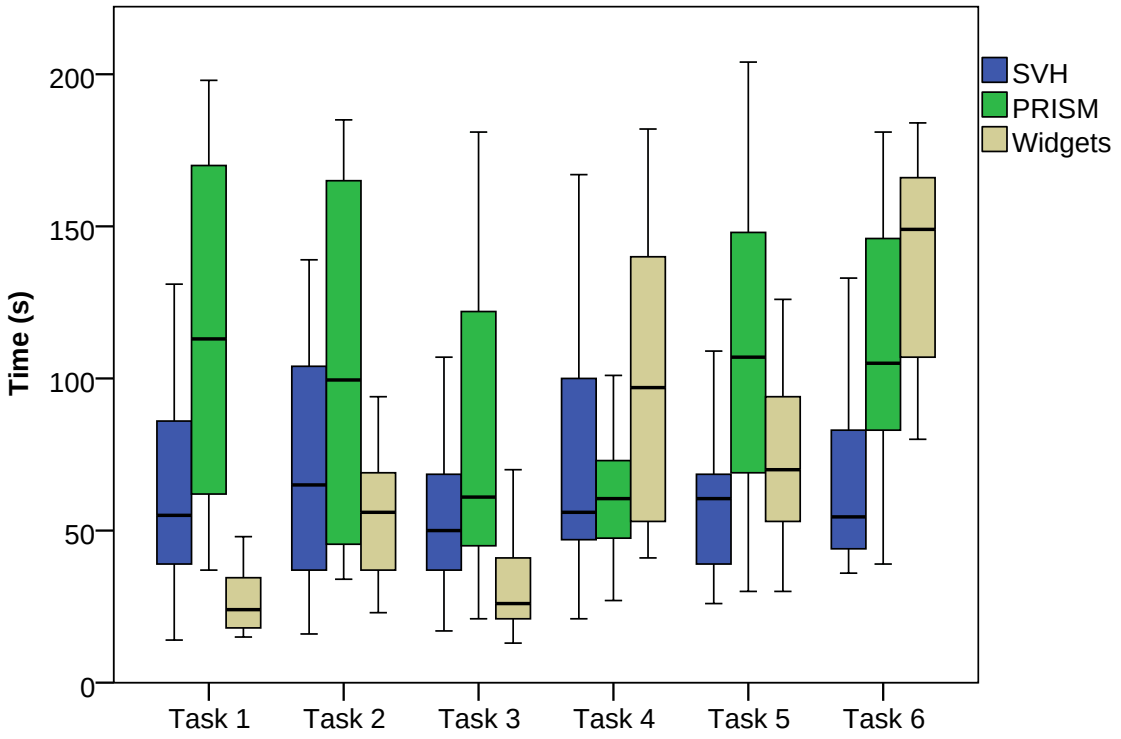


Figure 4.9: Time to complete the six tasks using the three techniques, in seconds. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

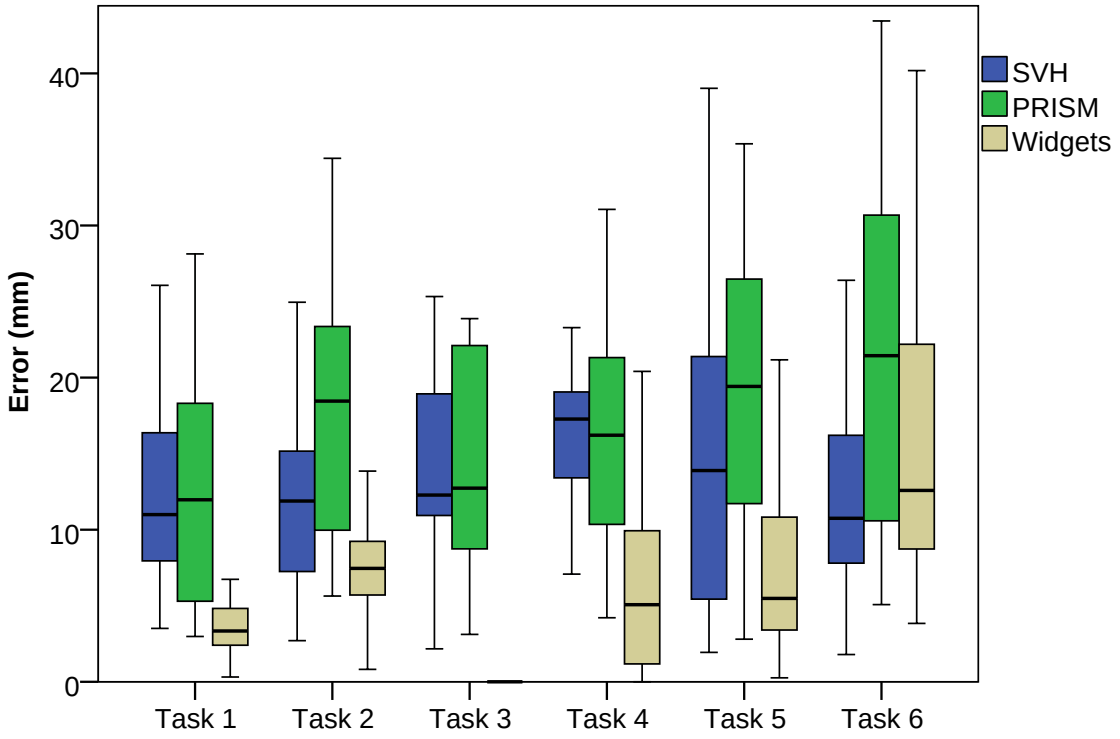


Figure 4.10: Position error attained in the six tasks using the three techniques, in millimeters. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

average: 9.8º) and PRISM (Task 1 average: 7.3º, Z = -3.408, p = 0.003; Task 2 average: 7.1º, Z = -2.803, p = 0.015).

Widgets might have outperformed both SVH and PRISM in the first task, due to its DOF separation. Since this task required translating the object along a single axis, the ability to manipulate with such constraint allowed users to avoid unexpected rotations and translations, thus preventing error. The same principle applies to time completion, because users did not need to correct mistakes. Similarly, the second task saw better results with Widgets in both translation and rotation error, although the time taken by users had no significant difference against SVH. We believe this occurred because transformation separation found in the Widgets technique made it impossible to take a direct path, requiring users to move along all three axes separately.

In the second pair of tasks we focused on rotations. Significant differences for execution time were only found for the third task ($\chi^2(2) = 20.985$, $p < 0.0005$), in which the use of Widgets (average: 27s) reduced time needed when compared to SVH (average: 53s, Z = -3.053, p = 0.006) and PRISM (average: 58s, Z = -3.823, $p < 0.0005$). For both tasks, position error revealed significant differences accord-

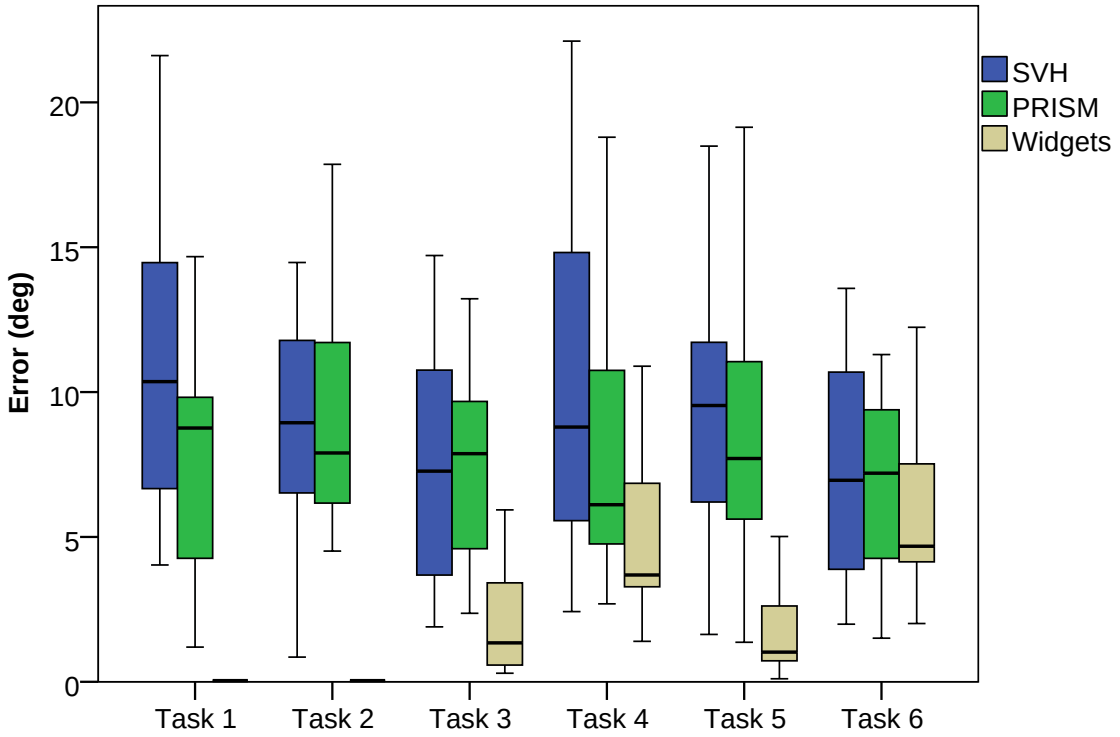


Figure 4.11: Rotation error attained in the six tasks using the three techniques, in degrees. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

ing to the technique used (Task 3: $\chi^2(2) = 16.545$, $p < 0.0005$; Task 4: $F(1.619, 14.575) = 6.586$, $p = 0.012$). Widgets (Task 3 average: 0.0mm, Task 4 average: 9.7mm) led to better positioning than SVH in both tasks (Task 3 average: 13.3mm, $Z = -3.296$, $p = 0.003$; Task 4 average: 15.7mm, $p = 0.008$) and than PRISM in the third task (average: 16.6mm, $Z = -3.059$, $p = 0.006$). Rotation error was also significantly affected by the techniques (Task 3: $\chi^2(2) = 20.118$, $p < 0.0005$, Task 4: $\chi^2(2) = 16.545$, $p < 0.0005$). Once again, Widgets (Task 3 average: 1.8^{o}, Task 4 average: 5.3^{o}) performed better than SVH in both tasks (Task 3 average: 8.8^{o}, $Z = -3.547$, $p < 0.0005$; Task 4 average: 8.7^{o}, $Z = -2.868$, $p = 0.012$) and than PRISM in the third task (average: 7.1^{o}, $Z = -3.574$, $p < 0.0005$).

Alike the first pair, third and fourth tasks revealed advantageous results for Widgets in both translation and rotation error. Even though the focus of these tasks shifted from translation to rotation only, the ability to separate transformations proved to be, once again, significant. The increased completion time found in the fourth task was a consequence of rotations around all axes. Users felt confused and unable to easily figure out the necessary rotations to reach the desired orientation.

The last pair of tasks required both translations and rotations. In both cases, techniques had an effect on the time participants took to complete tasks (Task 5: $F(1.422, 27.021) = 12.645$, $p < 0.0005$; Task 6: $\chi^2(2) = 27.900$, $p < 0.0005$). While in the fifth task PRISM (average: 102s) was outperformed by both Widgets (average: 72s, $p = 0.004$) and SVH (average: 63s, $p = 0.003$), in the sixth Widgets (average: 135s) took longer than SVH (average: 55s, $Z = -3.920$, $p < 0.0005$) and PRISM (average: 112s, $Z = -2.520$, $p = 0.036$). SVH was also faster than PRISM in the final task ($Z = -3.323$, $p = 0.003$). In both tasks, there were differences regarding error in object positioning (Task 5: $\chi^2(2) = 8.533$, $p = 0.014$, Task 6: $F(1.671, 23.391) = 5.232$, $p = 0.017$). Widgets (average: 6.6mm) reduced distance to target in the fifth task when compared to SVH (average: 15.1mm, $Z = -2.809$, $p = 0.015$) and PRISM (average: 21.4mm, $Z = -3.010$, $p = 0.009$). In the last task, SVH (average: 11.4mm) allowed users to place the object closer to its target position than PRISM (average: 21.2mm, $p = 0.048$). Analyzing rotation error, we only found significant differences in the fifth task ($\chi^2(2) = 22.625$, $p < 0.0005$), in which Widgets (average: 1.1^{o}) attained better results than SVH (average: 9.2^{o}, $Z = -3.823$, $p < 0.0005$) and PRISM (average: 8.9^{o}, $Z = -3.464$, $p = 0.003$).

Final tasks had an increase in complexity, since they both required participants to apply translations and rotations to the object. The time participants took to complete these tasks was negatively affected due to the necessary increased number of operations. As a consequence, translation and rotation error presented worse results when compared to previous tasks, because the time limit prevented participants to make final adjustments.

It is also worth of notice that both SVH and PRISM did not have major variations along all tasks, with no regard to its difficulty. For these techniques, after grabbing an object all tasks are alike, since there is no constraint in transformations being applied to the object. Taking the first and last task as an example, we used a Paired-Samples T-Test and no significant differences were found in time, translation error or rotation error. Moreover, PRISM and SVH consistently shared similar results. As PRISM's original authors pointed out [43], PRISM rotations are confusing for some users, which might have had a negative impact in tasks overall performance.

4.1.3.2. User Preferences

Using questionnaires, we asked the participants how they felt about each technique. This included general ease of use, translation and rotation difficulty, and fun factor.

	SVH	PRISM	Widgets
Easiness *	4 (1)	2 (1)	4 (1)
Translation	4 (2)	4 (1)	4 (1)
Rotation *	3 (2)	2 (1)	4 (2)
Fun *	3 (2)	2 (1)	4 (2)

Table 4.1: Participants preference for each technique, regarding different criteria (Median, Interquartile Range). * indicates statistical significance.

Participants were given a Likert scale from 1 to 5 to answer our questions, being 5 the favorable value. Answers are depicted in Table 4.1.

Analyzing attained results, we identified significant differences in ease of use ($\chi^2(2) = 19.547$, $p < 0.0005$), rotation difficulty ($\chi^2(2) = 25.352$, $p < 0.0005$) and fun factor ($\chi^2(2) = 13.216$, $p = 0.001$). Participants strongly agreed that PRISM was generally the hardest (Widgets: Z = -3.716, $p < 0.0005$, SVH: Z = -3.157, $p = 0.006$) and the least fun to use (Widgets: Z = -3.057, $p = 0.006$, SVH: Z = -2.463, $p = 0.042$). Widgets appealed more to participants to perform object rotations than SVH (Z = -2.863, $p = 0.012$) and PRISM (Z = -3.874, $p < 0.0005$). Also, participants agreed that it is easier to rotate objects using SVH than PRISM (Z = -2.708, $p = 0.021$). There was no difference in translation difficulty, even though PRISM sacrifices directness and time over enhanced precision. The Widgets approach, although requiring more effort for complex movements, was as appealing to participants as the other techniques. It is as fun as the direct manipulation approach, but with increased final placement.

4.1.3.3. Observations

Users found the widget-based approach as easy-to-use as SVH, and easier than PRISM. Overall, error attained in object placement using Widgets was smaller than with other approaches. However, this increased positioning sacrificed speed in more complex tasks. The results between SVH and PRISM are similar to the 6-DOF task from original PRISM evaluation [43], as we did not impose any minimum requirements for distance or angle between the object and its target placement. Indeed, in our evaluation, PRISM's translation was praised by participants, as its operation was easy to understand and its benefits were clear. The main issue with this

technique was found on rotations, where some users complained about it being confusing, as previously stated [43]. Since there is no complete DOF separation in PRISM and none in SVH, as opposed to the Widgets approach, extra hand tremor or tracker noise occasionally caused unwanted transformations. This was mostly noted when users desired to only translate the object and an accidental rotation occurred. Distinctly from SVH, users found it difficult to return to the correct orientation with PRISM when this disturbance was too strong, which had a severe impact on performance of all tasks.

Since our tracking solution considerably differs from that used in [43], we experimented with different values for PRISM's scaling constant hoping to find better suited ones. However, we ended using those originally proposed, as mentioned in Section 4.1.1.2. Because this constant simultaneously affects when scaling is applied and how much movement is scaled, we could not identify a better compromise. We also found the method used to calculate hand's speed very prone to be negatively impacted by tracker noise. Instead of using information from two consecutive frames, it uses the difference between the current hand's position and that from 500ms ago. However, this does not totally prevent noise, but potentially reveals it half a second later.

4.1.4. Lessons Learned

In this study, we tested our first hypothesis, by assessing the benefits of DOF separation in mid-air to reduce placement error over a direct approach, after it has been proved useful in other interaction paradigms by previous research. We compared three mid-air object manipulation techniques through a user evaluation: the direct Simple Virtual Hand [73], PRISM's scaled users' movements [43], and our implementation of mid-air virtual handles for single DOF control. We concluded that indeed the DOF separation through virtual widgets led to error reduction, at the cost of increased time for more complex tasks, thus verifying our hypothesis.

Also as a result of our evaluation, we were able to draw some tentative guidelines for object manipulation in mid-air:

- Direct 6-DOF manipulation is well suited for coarse transformations. It allows fast and natural interactions, although not offering accurate placement;
- It should be possible to perform translation and rotation operations independently. We found that, in both the Simple Virtual Hand and PRISM,

unwanted transformations happen when a simple translation or rotation is in order, which negatively impacts performance;

- Single DOF separation is very desirable for precise transformations, typically for fine-grain adjustments. This separation, more than separating translation and rotation, constrains transformations to a single dimension, preventing additional unwanted actions;

- Scaled transformations, as proposed in PRISM, are appealing only for translation. Scaled rotation confused participants, but they found separated scaled translations in each coordinate axis to be helpful in improving accuracy. Combining scaled translations with other approaches might improve their overall performance.

4.2. Combining DOF Separation with Scaled Movements

In the study of the previous section, we concluded that DOF separation can benefit mid-air manipulations, increasing placement precision. However, this is achieved while sacrificing completion time due to the additional transformations required for more demanding tasks. Following these results, we developed a novel technique, WISDOM, by compiling the best characteristics of the techniques evaluated into a single one. With this technique, we aim at offering mid-air manipulations that can be as precise as those attained with single DOF control, while reducing the time needed to do so, getting closer to the fast performance of direct approaches.

In this section, we begin by detailing WISDOM and all the different combinations of transformations and controlled DOF it offers. To validate its performance, we conducted a user evaluation, where we compared it against the two approaches that performed better in the previous study. We describe the method, tasks and apparatus used, as well as the people who volunteered to participate. Lastly, we report and discuss evaluation's results and lessons learned.

4.2.1. Proposed Technique: WISDOM

Gathering the best aspects of the techniques previously evaluated, we propose a novel mid-air manipulation technique: WISDOM (Widgets combining Scaled movements and DOF separation for Object Manipulation). It offers: direct manipulation for coarse manipulations, DOF separation to prevent unwanted transformations, and scaled movement for improved accuracy.

With WISDOM, users can switch between different manipulation modes by enabling or disabling widgets at any moment. Widgets are enabled and disabled by grabbing the object with extra pressure and quickly releasing. To achieve this additional level of pressure, users are required to apply considerably more strength than that required for common manipulation approaches, such as pressing triggers on hand-held controllers. This prevents unwanted activations.

Direct object manipulation is set off by grabbing the desired object when widgets are disabled. A similar action while widgets are enabled triggers isolated translation, with scaled movements similar to PRISM [42]. Scaled movements are also applied during widget translations, allowing for fine-grained adjustments after users lock the desired axis. In addition, we implemented isolated 2D TRS, and both uniform and non-uniform object scaling, a feature that is not common in mid-air techniques. Exploring two-handed interactions, users can trigger scaling and TRS by simultaneously grabbing two handles from the same axis or from different axes, respectively.

4.2.1.1. Direct manipulation

In our previous evaluation, we observed that tasks which required users to perform a larger number of operations had an increased completion time using our widgets implementation in comparison to the other techniques. These results, together with user feedback, suggest that the combination of direct manipulation with widgets might lead to a decrease in completion time in complex tasks. As such, WISDOM features 6-DOF direct manipulation. Users can activate it by directly grabbing the object when widgets are disabled.

4.2.1.2. Uniform Scaling

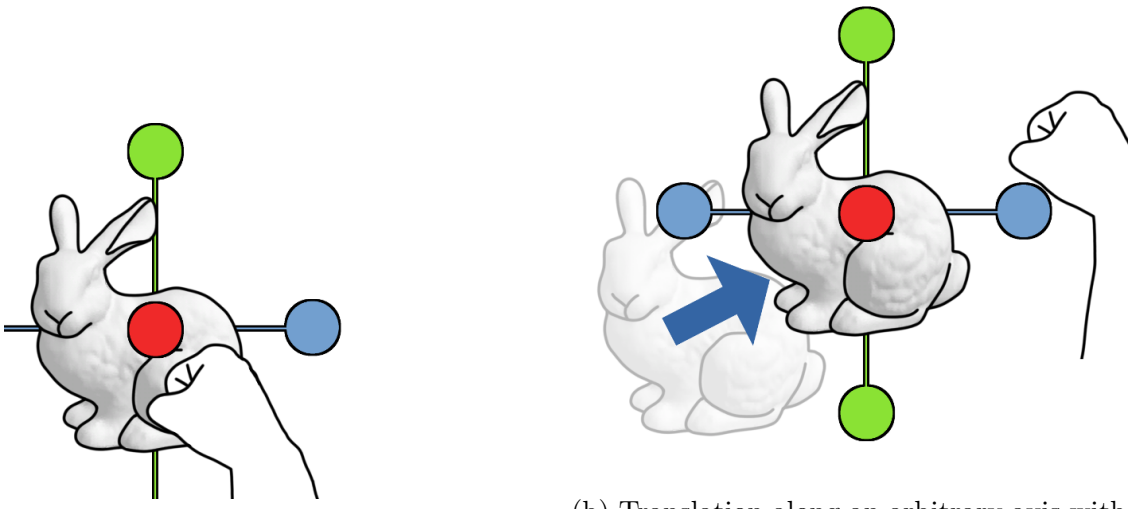
In order to perform uniform scaling with WISDOM, users need to grab directly the object with one hand, triggering direct manipulation, and then grab anywhere outside the object with the other hand simultaneously, similarly to the 6-DOF Hand technique described in Section 3.1.1.1. Any increase or decrease in the distance between both hands is used to scale the object up or down respectively. For this, we calculate the ratio between the current and previous hands' distance, which is then multiplied by the current object scale.

4.2.1.3. Widget-based Manipulations

Widgets presented significant advantages during user testing over the remaining techniques. When enabled, users have at their disposal all transformations offered by our previous widgets implementation, detailed in Section 4.1.1.3. Additionally, in WISDOM we implemented three more transformations: 3-DOF translation, non-uniform scaling and 2D TRS.

Translation

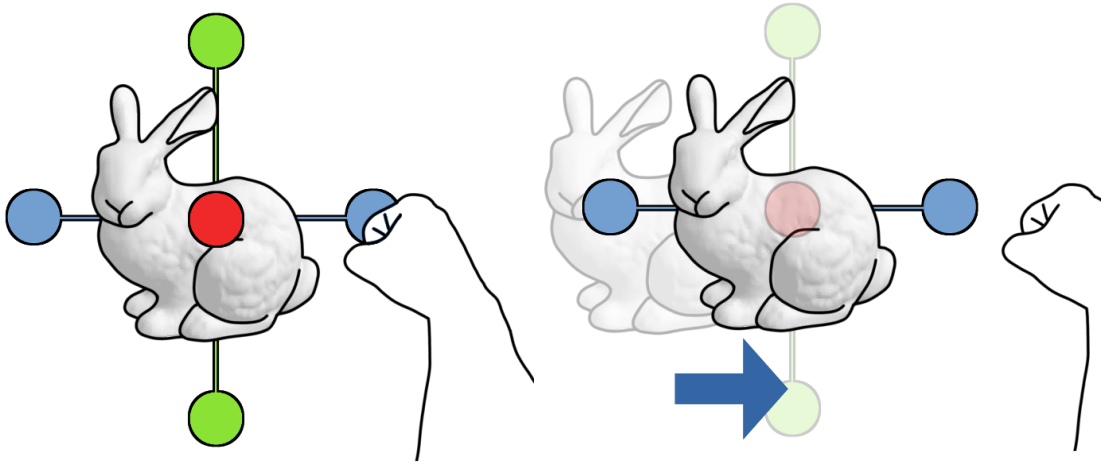
Users can enable widgets to translate in two ways: the first is triggered by grabbing the object directly and allows for 3-DOF translation, and the second makes use of



(a) Grab object while showing widgets.

(b) Translation along an arbitrary axis with accumulated offset.

Figure 4.12: Isolated scaled 3DOF translation.



(a) Grab a single axis handle.

(b) Translation along an horizontal axis with accumulated offset.

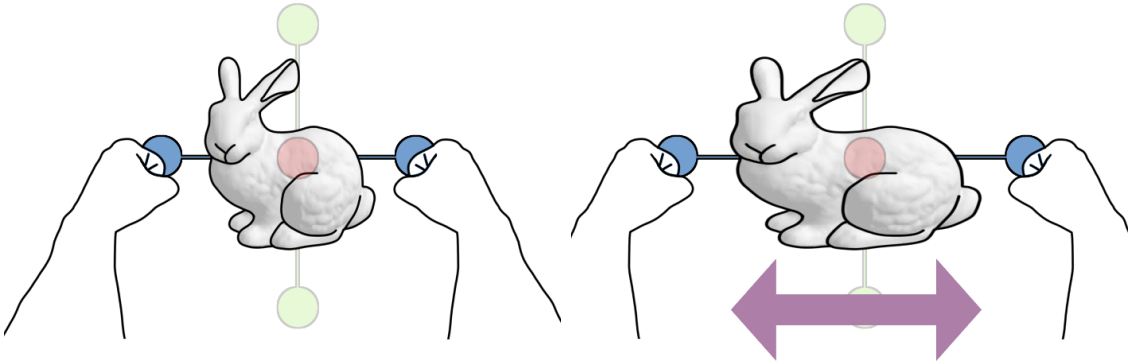
Figure 4.13: Widgets scaled translation.

widgets' handles to translate in 1-DOF. In 3-DOF translations, users are able to move the object in any direction with the benefits found in PRISM's scaled movements [43] (Fig. 4.12). However, instead of using only two hand position samples separated by 500ms, as it is done in PRISM, we calculate the weighted average of hand's speed between all frames in the same interval. With this approach, we expect to achieve smoother results while maintaining it frame rate independent.

Isolated 1-DOF translation was already provided in our widgets implementation. By grabbing the desired handle and dragging it along the axis, the object would lock to translations on that single axis. We kept this behaviour but added scaled movement, to perform translations with additional accuracy (Fig. 4.13). We believe that this approach might lead to improvements when fine-grained adjustments are needed.

Rotation

Rotation transformations were kept equal to our first implementation. Users can perform them by grabbing an handle and dragging it around one of the other handles' axes. The object will rotate in 1-DOF, following users' hand while locked to the defined axis.



(a) Grab two handles from the same axis. (b) Scale along the axis.

Figure 4.14: Non-uniform scaling.

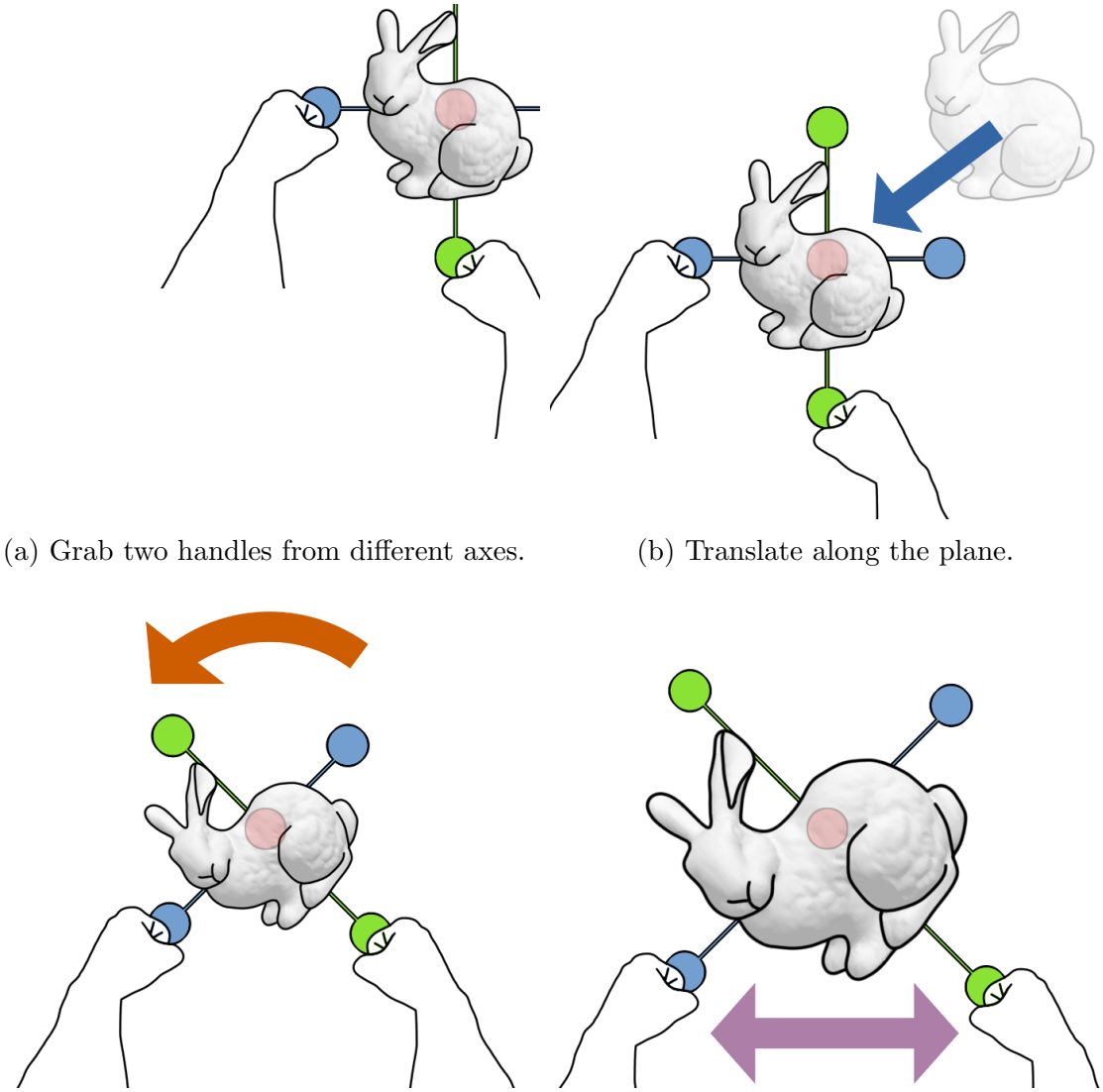
Non-uniform Scaling

Additionally to uniform scaling, WISDOM also provides non-uniform scaling. Relying on two handed interactions, users are required to grab both handles from the same axis. Moving both hands apart scales the object up while moving them together scales it down along that axis (Fig. 4.14). WISDOM also provides non-uniform 2-DOF scaling in the form of 2D TRS as following described.

2D TRS

We implemented a 2D Translate-Rotate-Scale (TRS) mode, or two-point rotation and scale [57], which is enabled by grabbing two handles from different axes (Fig. 4.15a). This technique is now the *de facto* standard for 2D manipulations on touch-enabled surfaces. We implemented this technique as it offers more freedom than 1-DOF manipulations while maintaining some constraints. Although it allows all three transformations to be applied simultaneously, they are restricted to the plane defined by both selected axes. Metaphorically, this plane can be seen as a table or a wall, with the corresponding physical constraints.

For translations, we calculate the difference between the current and last hands' mid-point and move the object in the resulting distance and direction (Fig. 4.15b). For rotations, we calculate the angle between the current and the last vector between both hands, and the rotation axis is defined by the plane normal (Fig. 4.15c). Regarding scaling, we use the distance between both hands similarly to previously



(a) Grab two handles from different axes.

(b) Translate along the plane.

(c) Rotate around the plane normal.

(d) Scale along both axes.

Figure 4.15: 2D TRS.

described scaling transformations, but only apply this scaling to selected axes, as depicted in Figure 4.15d.

4.2.2. User Evaluation

To validate WISDOM, we followed a similar methodology to the previous evaluation. We compared it against two techniques through a set of tasks in the same environment.

4.2.2.1. Baseline Techniques

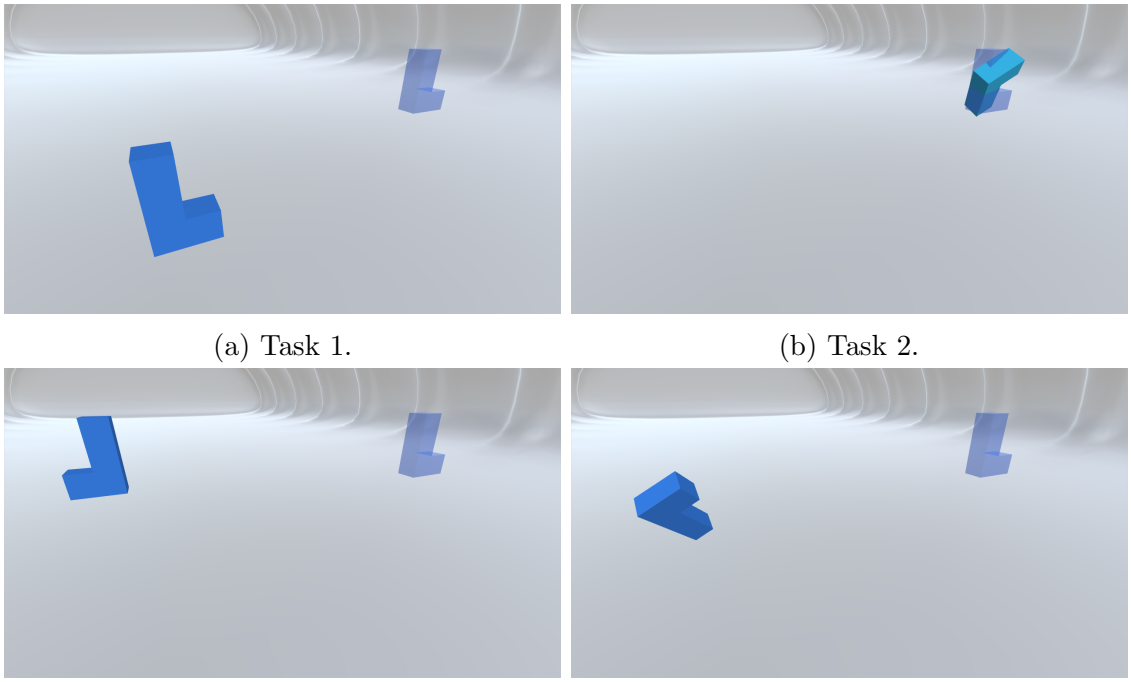
As baselines, we chose the Simple Virtual Hand [73] (SVH) and the Widgets techniques. Those achieved the best results in the previous study, and each one has its different strengths and weaknesses. Our aim is to assess if WISDOM can achieve the same precision as Widgets, while being as fast as SVH. Since WISDOM is the only technique supporting scaling transformations, and to focus our study, we focused on positioning tasks and disabled scaling in all its implementations. WISDOM's 2D TRS mode was kept enabled, although only allowing for translation and rotation.

4.2.2.2. Procedure

The same structure was followed across all user sessions, each lasting approximately 40 minutes. The experiment was performed in our laboratory, with a controlled environment. We began by introducing to the participants the experiment they were about to perform, followed by a brief description of the techniques being evaluated. Again, the techniques were performed in alternated order, following a Latin square design, assuring that each one was experienced in every possible permutation, in order to avoid biased results. We played a video showing how to apply transformations to the object with each technique. After the video, participants had a training period of three minutes, or less if they considered themselves to be already acquainted, to explore the approach in a dedicated environment. After completing each technique's tasks, participants fulfilled a questionnaire to classify the technique according to several aspects. The experiment concluded with a questionnaire to profile the participants.

4.2.2.3. Tasks

Participants were asked to complete a set of four tasks for each technique. These were a subset of our previous evaluation tasks. Since it was already shown that for 1-DOF tasks Widgets was both the fastest and more accurate, we focused on more the complex tasks, with 3 and 6-DOF. All consisted in a docking task, where participants had to place a L-shaped block in the same position and orientation of a transparent copy. Each task was limited to a maximum of three minutes in order to prevent



(a) Task 1. (b) Task 2.

(c) Task 3. (d) Task 4.

Figure 4.16: Tasks performed by the participants.

excessively long sessions. After reaching the time limit we informed participants they could stop, and we considered the attained position and orientation as final.

For the first task (Figure 4.16a), the object to be manipulated begun with the correct orientation, but with a wrong position along all three coordinates. The second task (Figure 4.16b) implied only rotation around an arbitrary axis, as the object was already in the correct position. In the third task (Figure 4.16c), users were required to rotate the object around the YY axis and translated along both XX and ZZ axes. Finally, in the last task (Figure 4.16d), participants had to apply full 6-DOF transformations to the object. Although some tasks required only one kind of transformation, none was restricted.

4.2.2.4. Setup and Prototype

For this evaluation, we used the same hardware setup from the previous study, presented in Section 4.1.2.3, with a single addition. Since we only had one hand tracker device, we coupled it with a Genius Ring Mouse, shown in Figure 4.17. It was placed on participants' non-dominant hand, to detect grab gestures for two-handed TRS transformations in WISDOM. Also, the pressure pad on the dominant hand was used this time not only to detect grabbing actions, but also to identify

Figure 4.17: Genius Ring Mouse used for discrete input.

two different pressure levels. The second level required considerably more strength than usual grabs, and was used to activate and deactivate the virtual widgets. The virtual environment was kept from the previous evaluation, and the L-shaped block was the only virtual object that could be grabbed and transformed.

4.2.2.5. Participants

We counted with the participation of 20 people (three female), between the ages of 18 and 40 years old, with the majority (65%) between 18 and 25. Most had at least a BSc degree (80%), while the remainder are finishing it. More than half (60%) had never experienced a Virtual Reality setting, and 45% had never used any kind of gesture recognition systems, such as Microsoft Kinect, Wii Remote or Playstation Move. 25% of participants reported using 3D modelling systems at least once a month.

4.2.3. Results and Discussion

We collected both objective data during our experiment, through logging mechanisms, and subjective data, asking participants to fill out questionnaires. We used Shapiro-Wilk test to assess data normality. We then ran the repeated measures

ANOVA test to find significant differences in normal distributed data, and Friedman non-parametric test with Wilcoxon Signed-Ranks post-hoc test otherwise. In both cases, post-hoc tests used Bonferroni correction (presented sig. values are corrected).

4.2.3.1. Task Performance

We measured time taken by participants to fulfill each task, as well as object placement error. Time taken for all tasks, in seconds, is depicted in the graph of Figure 4.18. Regarding errors, we registered both position error, in millimeters (Figure 4.19), and rotation error, in degrees (Figure 4.20).

For the translation only and rotation only tasks, we found statistically significant differences in completion time (Task 1: $\chi^2(2) = 14.282$, p = 0.001; Task 2: $\chi^2(2) = 14.282$, p= 0.001). For the first task, post-hoc tests revealed SVH (average: 53.7s) to be faster than both Widgets (average: 108.4s, Z = -3.310, p = 0.003) and WISDOM (average: 117.8s, Z = -3.354, p = 0.003). In the second task, SVH (average: 49.1s) was also faster than Widgets (average: 107s, Z = -3.332, p = 0.003)

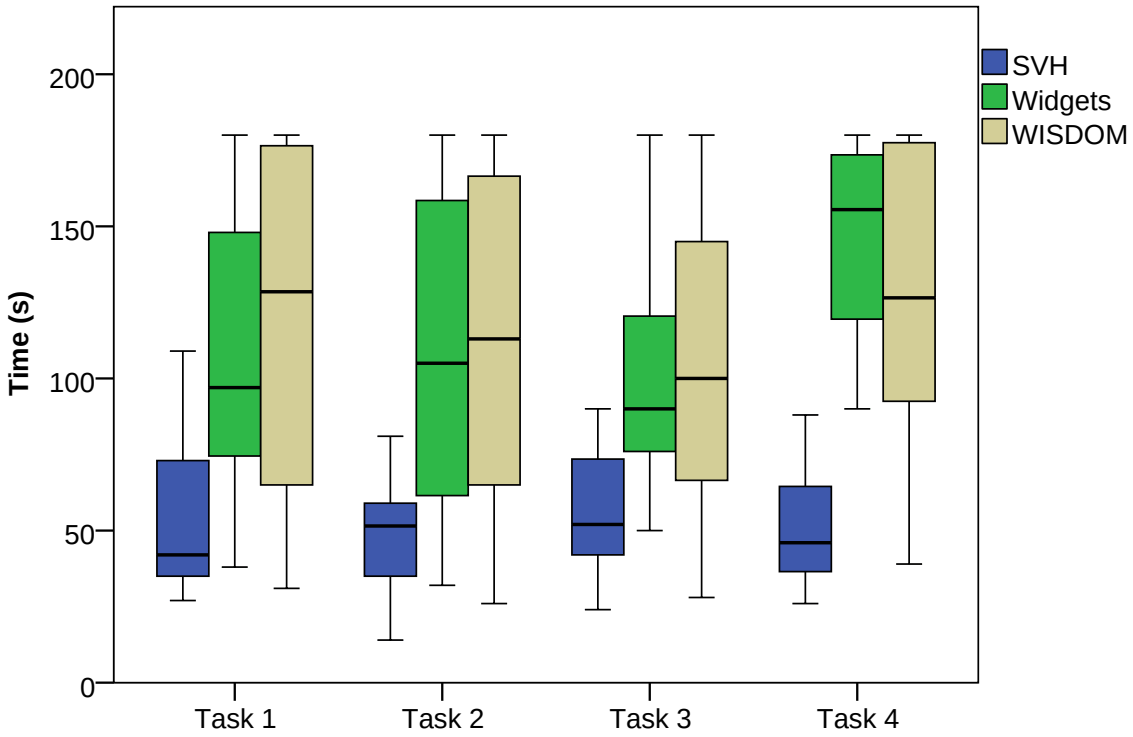


Figure 4.18: Time to complete the four tasks using the three techniques, in seconds. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

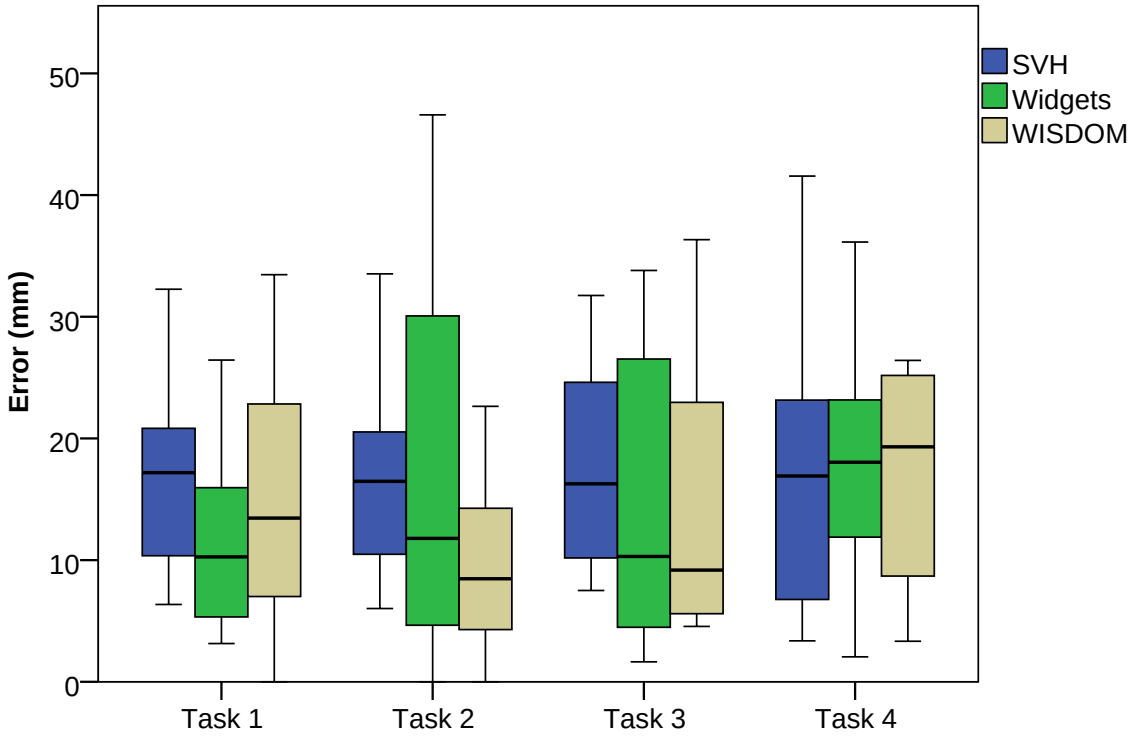


Figure 4.19: Position error attained in the four tasks using the three techniques, in millimeters. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

and WISDOM (average: 113.5s, Z = -3.332, p = 0.003). The technique used also influenced rotation error in the first task ($\chi^2(2) = 11.804$, p = 0.003), with SVH (average: 5.8º) performing worse than Widgets (average: 3º, Z = -2.534, p = 0.033) and WISDOM (average: 3.1º, Z = -2.900, p = 0.012), and position error in the second task ($\chi^2(2) = 7.964$, p = 0.019) with WISDOM (average: 9.6mm) outperforming SVH (average: 16.3mm, Z = -2.605, p = 0.027).

Even though SVH was the fastest approach in both tasks, it didn't achieve the same level of precision regarding translation and rotation error. The first task, which could be completed without applying any rotation to the object, showed that separating transformations benefited both Widgets and WISDOM. The same principle applies to the second task, where users were required to rotate the object around an arbitrary axis. Translations were inevitable due to the nature of SVH as opposed to the possibility of separating translation from rotation in WISDOM.

The second pair of tasks required both translations and rotations. In both cases, techniques had an effect on the time participants took to complete tasks (Task 3: $\chi^2(2) = 18.000$, p < 0.0005; Task 4: $\chi^2(2) = 24.108$, p < 0.0005). SVH (average: 56.6s) outperformed Widgets (Task 3 average: 103.4s, Z = -3.623, p < 0.0005;

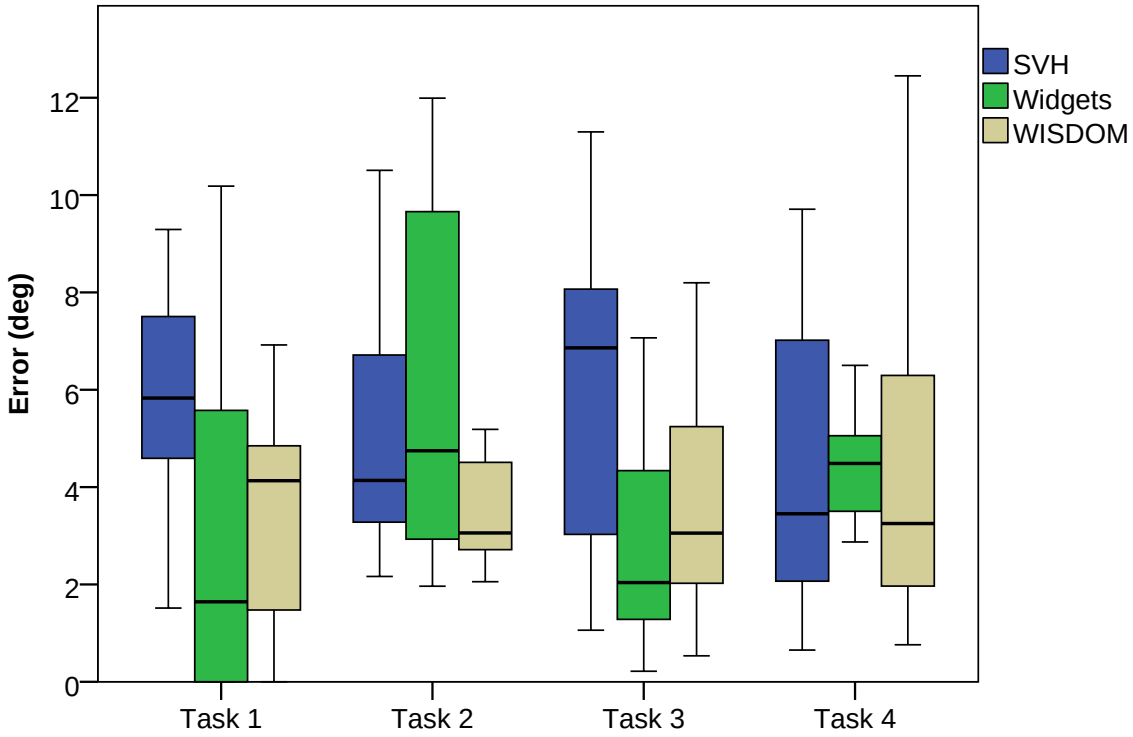


Figure 4.20: Rotation error attained in the four tasks using the three techniques, in degrees. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

Task 4 average: 146.2s, Z = -3.824, p < 0.0005) and WISDOM (Task 3 average: 106.8s, Z = -3.260, p = 0.003; Task 4 average: 124s, Z = -3.623, p < 0.0005) in both tasks. There were differences regarding error in object orientation in the third task (F(1.722, 25.829) = 14.436, p < 0.0005), where SVH (average: 6^{o}) was outperformed by both Widgets (average: 2.7^{o}, p = 0.001) and WISDOM (average: 3.6^{o}, p = 0.007).

Final tasks' increased complexity benefited SVH. The number of operations necessary to complete the tasks in both Widgets and WISDOM lead to longer completion times, despite the latter offering the option to either apply object transformations with 6-DOF or widgets. In the third task, SVH presented less accurate rotational results compared to Widgets and WISDOM. This might be related once again to the inability to separate transformations with SVH.

4.2.3.2. User Preferences

We asked the participants how they felt about each technique using questionnaires. This included general easiness of use, translation/rotation difficulty and fun factor.

	SVH	Widgets	WISDOM
Easiness	4 (1)	4 (2)	4 (1)
Translation	4 (1)	3 (2)	4 (1)
Rotation	4 (2)	4 (2)	4 (1)
Fun	4 (2)	4 (1)	4 (1)

Table 4.2: Participants preference for each technique, regarding different criteria (Median, Interquartile Range).

Participants were given a Likert Scale from 1 to 5 to answer our questions, being 5 the favorable value. Answers are summarized in Table 4.2. Analyzing attained results, we identified no statistically significant differences in any of the preferences. Since none of the techniques achieved a median of five, there is room for improvements across every criteria for all techniques. Widgets produced a slightly lower result regarding translation difficulty, which might be related to the increased number of operations required in complex tasks. Despite combining the positive aspects from SVH and Widgets, WISDOM didn't present any significant improvements to participants.

4.2.3.3. Observations

We believed WISDOM would present low completion times by providing both direct and single DOF manipulation, while achieving a placement error at par with the Widgets approach. Our initial thought was that users would use 6-DOF manipulations to place the object close to the target and then enable widgets to increase placement accuracy.

We observed that with SVH participants were fast placing the object near the objective, and did not try very hard to be accurate. As expected, with WISDOM they tended do the same coarse placement initially, but then improved it using widgets. Indeed, Figure 4.21 shows that participants spent similar times using the direct and widgets modes in WISDOM during tasks' execution. Naturally, this lead to WISDOM being slower than SVH, while achieving additional precision. However, WISDOM did not get better placement errors than Widgets. This might be due to the initial coarse positioning with 6-DOF causing undesired transformations, which then took time to be corrected. This led to similar times with Widgets, that pre-

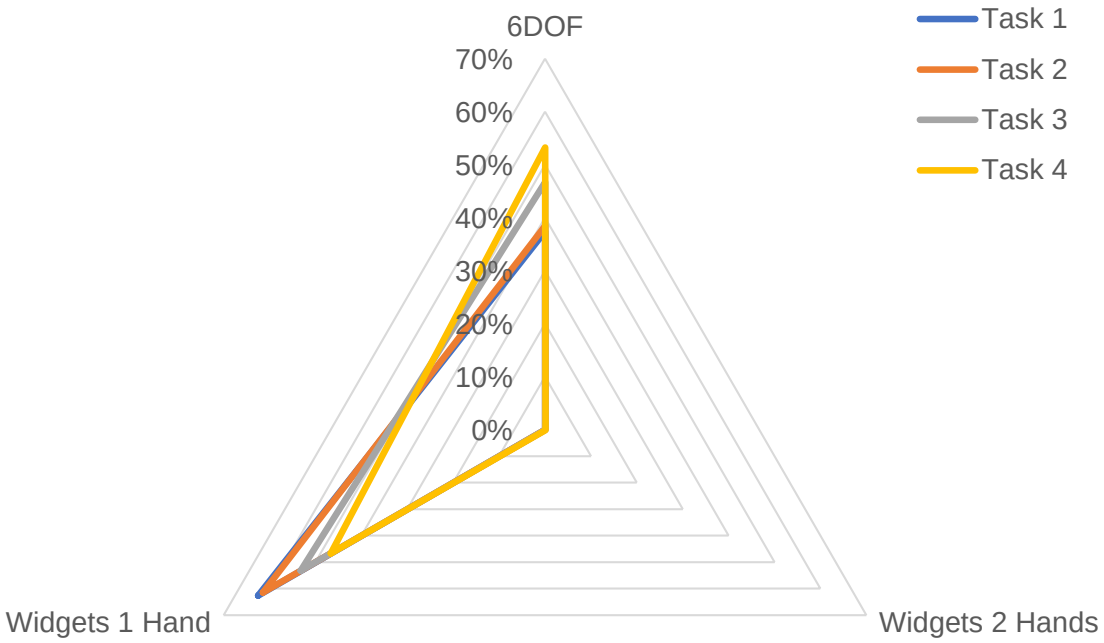


Figure 4.21: Time distribution in WISDOM, between the 6-DOF direct approach, Widgets 1 Hand (1-DOF and 3-DOF translation, and 1-DOF rotation) and Widgets 2 Hands (2D TRS).

vented those transformations, but required additional steps due to its single DOF transformations.

We also noted that participants preferred to use the 6-DOF mode for the initial tasks' steps instead of 3-DOF translation, which could have helped preventing some unwanted transformations. We believe this occurred for two reasons: isolated 3-DOF translations might not fit users mental model, since when they grab and drag an object they might expect it to behave similarly to the physical world, moving and rotating according to their hand, thus preferring 6-DOF; and the high number of available actions in WISDOM, which may have been too much, instigated participants to stick to a subset, namely 6-DOF and 1-DOF transformations. WISDOM also provided 2D TRS, which could be used at any time during tasks, and could be useful in the third task, allowing it to be done in one action. However, and similarly to 3-DOF translations, users forgot or didn't see the benefits of making use of it, as also shown in Figure 4.21.

4.2.4. Lessons Learned

While mid-air gestures in immersive virtual environments allow natural interactions, it is still difficult to place an object accurately, with the desired position and orientation. Following the guidelines drawn from the study of the previous section, we proposed a novel manipulation technique, WISDOM, which provides direct manipulation, as found in the Simple Virtual Hand [73] and the scaled movements used in PRISM [43], while maintaining DOF separation offered by our widgets implementation. Additionally, it supports 3-DOF translation, scaling and 2D TRS. Scaling and TRS explore two-handed interactions. To our knowledge, WISDOM is the first mid-air technique that provides both uniform and non-uniform 1-DOF and 2-DOF scaling in mid-air.

In order to validate WISDOM, we conducted a second user evaluation. WISDOM improved placement precision in some tasks, but it didn't reduce completion time, as opposed to our initial beliefs. This might be related to its comprehensive list of features, which participants may have felt difficulties in recalling on how to get the best of them.

When comparing the attained results with those from the previous DOF separation assessment, we found differences regarding completion time and placement error. This led us to confront those guidelines with these new results. The Simple Virtual Hand was significantly faster than Widgets in tasks requiring 3-DOF manipulations, as opposed to our initial evaluation. This substantiates our first guideline about direct 6-DOF manipulations being suited for fast and coarse transformations, and reveals once again the extra time required for 1-DOF manipulations according to fixed frames.

Widgets did not show the same significant results in placement error in tasks requiring rotations around an arbitrary axis (task 2) and combinations of 2-DOF translation with 1-DOF rotation (task 3), suggesting that our third guideline, which concerns single DOF control for fine-grained adjustments, might be only valid in certain circumstances, namely for tasks requiring only 1-DOF. On the other hand, WISDOM helped prevent unnecessary transformations, corroborating our second guideline regarding transformation separation.

WISDOM's scaled translation followed the fourth guideline, combining it with the widgets for single DOF manipulation, in order to test our second hypothesis. However, as it didn't see any significant benefits in terms of position error over the Widgets approach, we could not validate the hypothesis. While we still believe this

might be a good approach for more precise translations, an implicit approach as found in PRISM possibly isn't the best.

4.3. Using Custom Transformation Axes

As we have been discussing in this chapter, the spatial input typically associated with VR setups can offer natural metaphors, allowing users to grab, move and rotate objects in a similar way to how it is done in the physical world, but mid-air gestures compromise object placement accuracy, whether due to limitations in tracking solutions or human dexterity itself. We showed that the constraint of transformations through DOF separation in mid-air can significantly improve object placement accuracy. However, to perform several transformations on two or more axes, it is necessary to perform multiple consecutive operations.

Now, we explore custom transformation axes, as an alternative to the traditional fixed frames. We assess whether such approach combined with scaled movements allow users to achieve the same level of precision as single DOF manipulations, while minimizing the impact in the time required to perform more complex tasks. For this, we propose a new manipulation technique, MAiOR, that supports both direct 6-DOF manipulation and transformation separation, as well as single DOF manipulation according to user specified transformation axis. Is also allows users to switch between exact and scaled mappings for object translations.

In the remainder of this section, we start by describing MAiOR. We follow with a user evaluation, were we compare MAiOR against a direct 6-DOF approach and single DOF widgets. Then, we discuss evaluation's results and summarize the lessons learned from this study.

4.3.1. Proposed Technique: MAiOR

To explore custom transformation axis in mid-air object manipulations, we developed MAiOR (Mid-Air Objects on Rails). It offers transformation separation and single-DOF manipulation on custom axes, using a rail metaphor, as well as 6-DOF direct manipulation and scaled movements for fast and accurate transformations, respectively. Figure 4.22 shows how to activate available transformations and constraints in MAiOR.

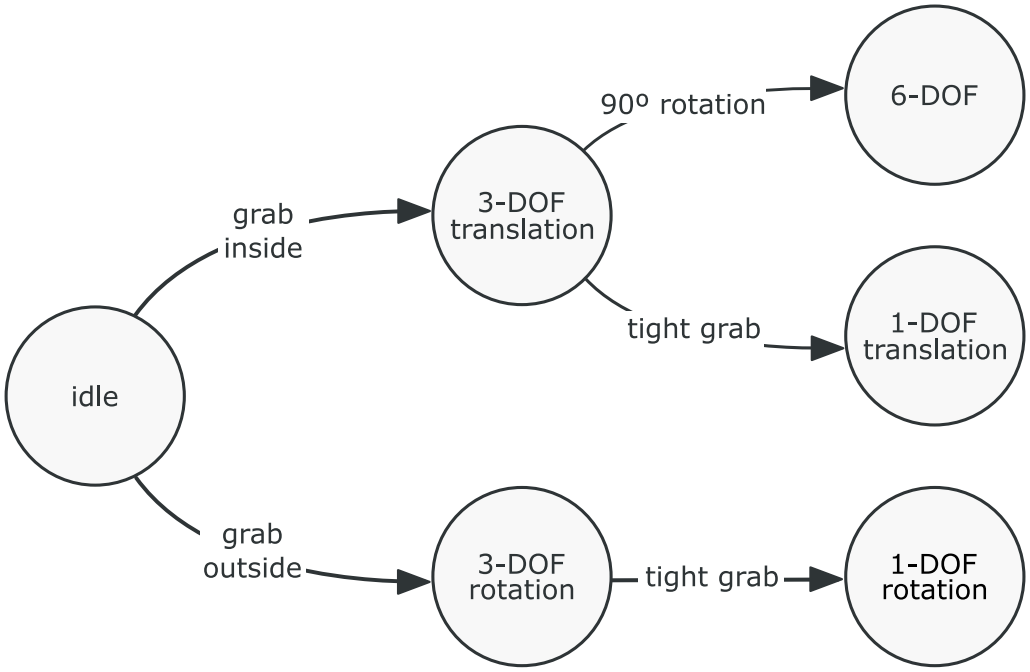


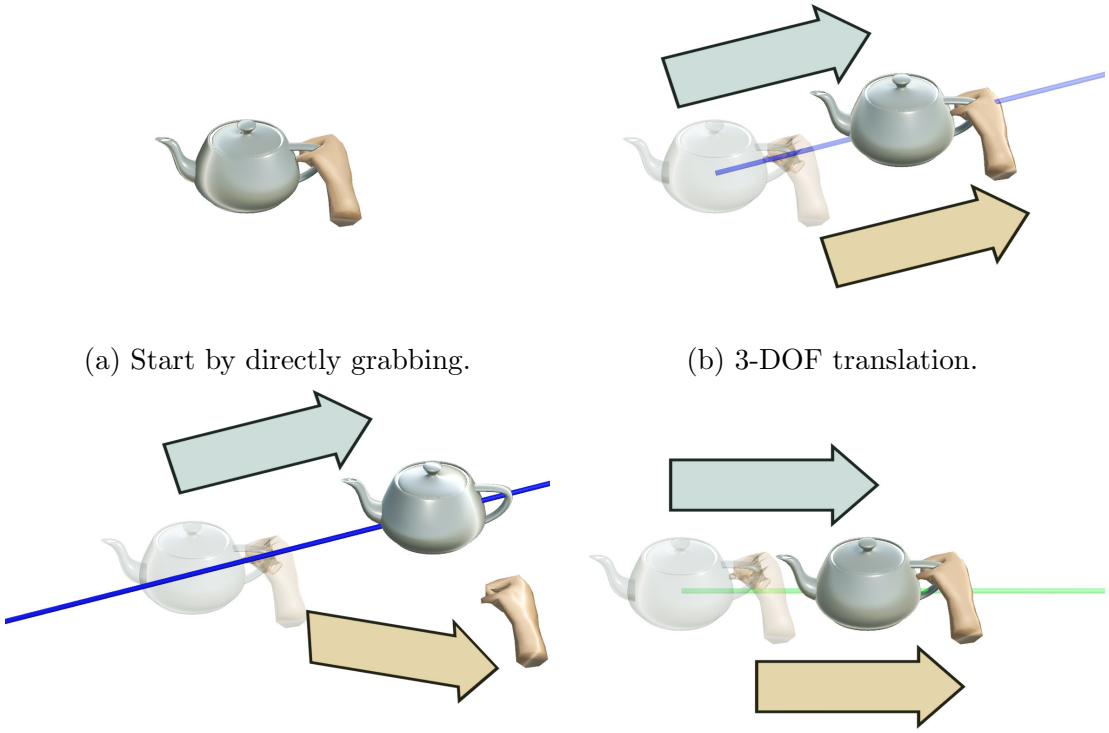
Figure 4.22: MAiOR's interaction diagram.

4.3.1.1. Translation

Translations can be performed in MAiOR by directly grabbing the desired object. Initially, the object will be restricted to 3-DOF translations. A transparent blue axis is drawn from the initial position of the object and passing through its current position (Figure 4.23). If the axis has a 10 degree or less deviation from any candidate axis, either from world or object reference frames, the closest candidate axis is shown instead, similarly to some approaches for interactive surfaces [109, 5], and its color is changed to green (world) or yellow (object).

The displayed axis can then be used to restrict translations to 1-DOF exclusively in that same axis. It can be locked at any time after starting a translation, by tightly closing the hand. When the axis is locked, it becomes opaque and infinitely extended for both sides. If a candidate axis has been selected, the object is re-positioned in such a way that the translation made since it was initially grabbed is coincident with the axis. Thereafter, variations in user's hand position will be projected on the selected axis, and object translations will follow the metaphor of an object on a rail.

Since scaled movements may improve precision in object placement when small adjustments are required, we implemented scaled translations using a fixed scale factor. This can be used both in 3-DOF and 1-DOF translations, by closing the non-



(a) Start by directly grabbing.

(b) 3-DOF translation.

(c) 1-DOF translation.

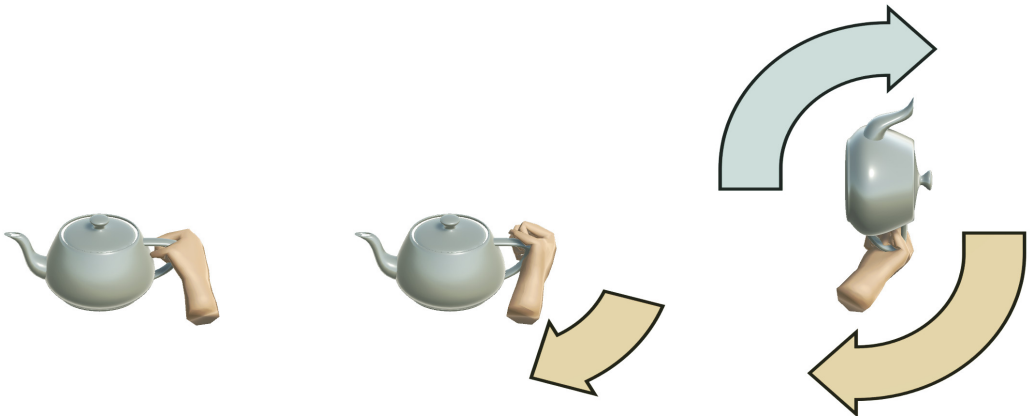
(d) Snap to candidate axis.

Figure 4.23: MAiOR's translation.

dominant hand. We chose a scaling ratio of 0.25. Naturally, this method originates the accumulation of an offset between the object and the hand.

4.3.1.2. Direct Manipulation

Since direct manipulation is the most efficient approach for coarse operations that require multiple transformations, MAiOR also supports it. While in translation mode, hand rotations are discarded until they achieve a 90° angle or greater in any direction (Figure 4.24), which triggers 6-DOF transformations. This gesture follows the metaphor of unlocking a door, because the object will no longer be locked to 3-DOF translations. Whenever this mode is activated, the object is immediately rotated by the same amount that the hand has rotated since the grab gesture, in order to ensure that the orientation of the object is consistent with the orientation of the hand.



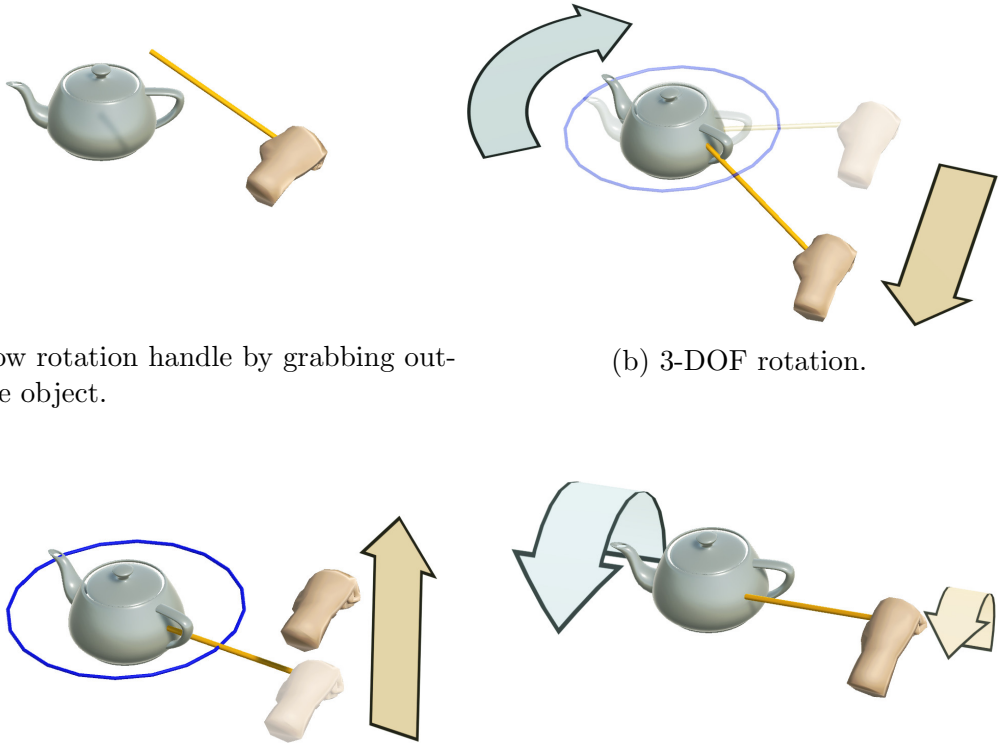
(a) Object grabbed. (b) Hand rotated 45°. (c) Hand rotated 90°.

Figure 4.24: Unlocking MAiOR's 6-DOF manipulation.

4.3.1.3. Rotation

Following the suggestions from Veit et al. [125], we also allow the decomposition of orientation tasks into single DOF manipulations. MAiOR's rotations are based on the same principles of DOF separation applied in translation, as depicted in Figure 4.25. Users start by rotating the object in 3-DOF and then can select an axis to rotate in 1-DOF. For this, users first have to close a hand outside the objects to enable a virtual bar (Figure 4.25). This bar will act as a lever for rotating objects. After attaching the bar to the desired object, the object can be rotated in 3-DOF. This approach is based on techniques such as the Handle-Bar [116] and the Spindle+Wheel [29], which have been shown to have good results in previous studies. However, in our approach we use the center of the object instead of a second hand, since the object remains in the same position, and this point is used as center of rotation. The rotation angle to be applied will be calculated according to the variation in the bar's orientation. This is, in turn, determined by the position of the user's hand, so that the intersection point between the bar with the object remains unchanged. In addition, we also implemented rotations around the axis defined by the bar, using wrist rotations.

After entering the 3-DOF rotation mode, a transparent blue circumference is shown around the object illustrating the current object rotation, taking into account hand position's variation (rotations around the wrist are not considered for this purpose). Similar to translation, if the calculated axis of rotation has a deviation of 10 degrees or less from a candidate axis, the circumference will be shown around it, either in green or yellow. Users can then lock the current rotation axis, by tightly closing



(a) Show rotation handle by grabbing outside the object.

(b) 3-DOF rotation.

(c) Motion parallel to locked rotation axis is discarded.

(d) Wrist rotation.

Figure 4.25: MAiOR's rotation.

the hand. Thereafter, to calculate rotation angle, users' hands movements will be projected in the plane defined by the circumference shown.

Since scaling rotations has a negative effect on users, as we saw in Section 4.1, we did not implemented an explicit precision mode in rotations. However, it is possible to scale rotations implicitly, due to the concepts behind circular motion. The further away the user's hand is from the object, the more distance will have to be covered for the object to rotate. This can be used to improve precision when fine-tuning object's orientation.

4.3.1.4. Scaling

Although not the mains focus for this study, MAiOR also provides scaling transformations, both uniform and constrained to a single axis. This approach is also based on the Handle-Bar [116], and uses two-handed input. With both hands closed, a virtual cursor is displayed at their midpoint. By colliding the cursor with the de-

sired object and tightly closing one hand, the cursor is transformed into a bar that adjusts itself dynamically to the distance between the hands. This bar illustrates the axis on which the object will be scaled, and can be changed by the users before starting the actual transformation. To confirm the scaling axis, users must tightly close the other hand. Thereafter, moving the hands away from one another increases the scale of the object, while moving the hands closer decreases it.

Similarly to the transformations described in the previous sections, if the chosen axis is sufficiently close to any of the candidate axis, it is converted to that axis and its color changed to yellow. Here, only object reference frame's axes are considered. Selecting the bar in such cases will activate non-uniform scaling, and it will be performed on the chosen axis. Otherwise, if none of the object frame's axes are suitable, the bar will turn blue and uniform scaling will be performed.

4.3.2. User Evaluation

To assess whether the custom transformation axis implemented in MAiOR appeal to users and help achieving an accurate and fast object placement, we conducted a user evaluation. We compared MAiOR against two baseline approaches, with a set of object placement tasks with different requirements.

4.3.2.1. Baseline Techniques

As baselines, we chose the two techniques that achieved the best results in the study of Section 4.1: the Simple Virtual Hand [73] and an indirect approach based on 3D widgets. The first was identified as the fastest, while the second was the most accurate. These are the same techniques used as baselines in the evaluation of Section 4.2, and we did not use WISDOM because it did not perform significantly better the others.

The Simple Virtual Hand (SVH) is often used as a baseline for evaluating other techniques, as it simulates as closely as possible interactions with physical objects. Manipulations start by directly grabbing the desired object. It will then closely follow the user's hand, moving and rotating accordingly. As both translation and rotation are applied simultaneously, there is no transformation separation. The point grabbed in the object will be the center of all transformations, until it is released.

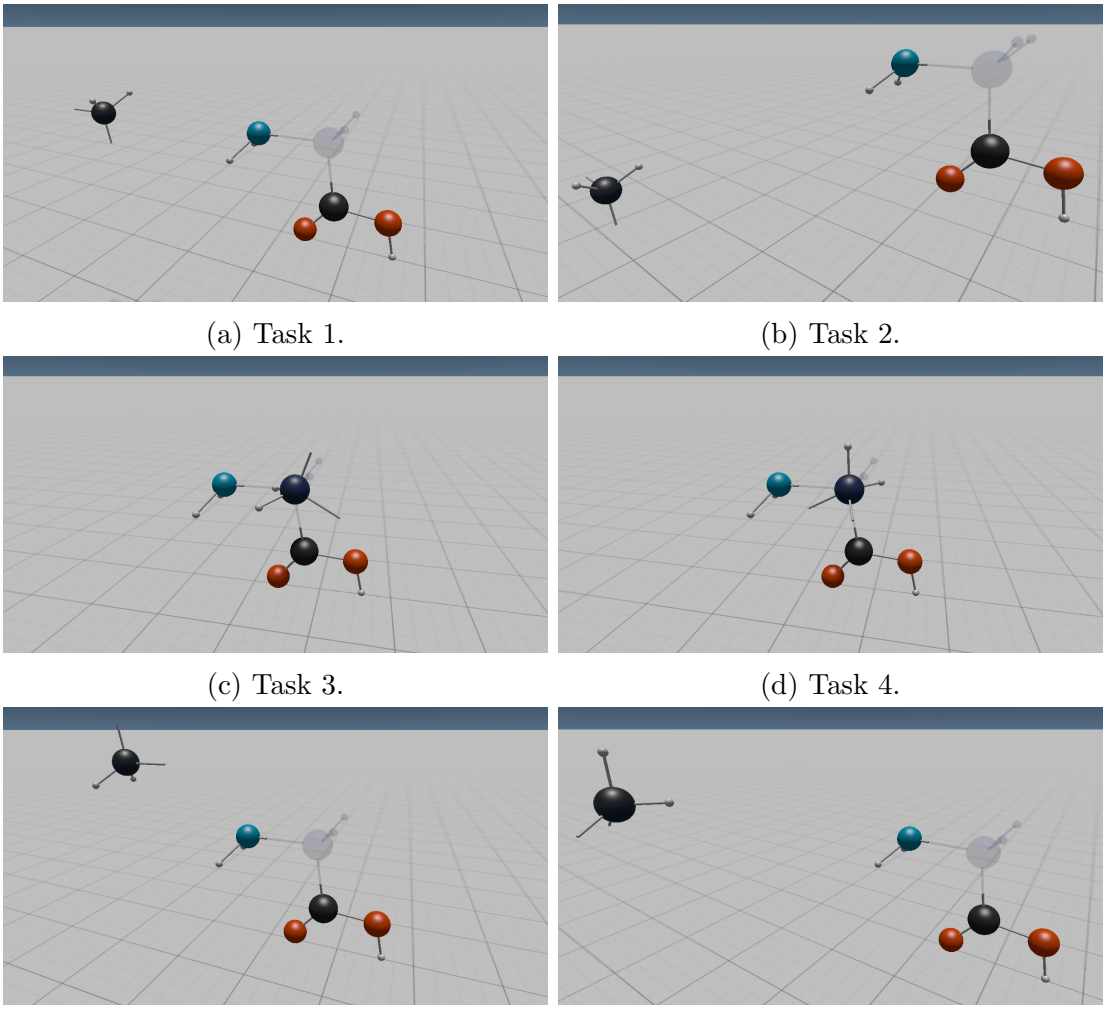
The Widgets technique is based on the 3D Virtual Handles common in mouse-based interfaces, as initially proposed by Conner et al. [32], which we adapted to mid-air interactions. It strictly follows explicit DOF separation, allowing only one transformation at a time according to a single axis form the object frame. The widget is composed by three cylinders representing object axes with spherical handles in each end, following a RGB coding for XYZ axes respectively. Translations can be done by grabbing an handle and moving the hand along the corresponding axis. Rotations are performed by also grabbing and handle, but rotating it about the desired axis. The decision to either perform a translation or rotation is made based on the first 10 cm from the hand's path after grabbing the handle. The transformation and axis resulting from that decision will remain locked until the handle is released.

4.3.2.2. Procedure

All sessions followed the same structure, and could last a maximum of 70 minutes. They were performed in our lab, which has restricted access, thus providing a calm and controlled environment. After an introduction to the evaluation, participants experimented all techniques. Techniques' order followed a Latin square design, to avoid biased results. For each technique, we started by playing a demo video explaining it, and gave a maximum of 5 minutes for participants to freely explore it and get acquainted with the environment. Then, we asked participants to execute a set of tasks and fulfill a questionnaire.

4.3.2.3. Tasks

Participants were asked to completed a set of six docking tasks for each technique. This set was based on those used in the previous user evaluations. The objective of all tasks was to place a carbon component on a model of a protein compound (Figure 4.26). The model was designed so that there was only one correct way to fit it. When the carbon component was placed on the docking model within the error boundaries, its color turned green and the task goal was achieved. To foster an highly accurate object placement, we set error boundaries to 1 millimeter for position, a rounded value of Frees and Kessler's very difficult task [42], and 1 degree for orientation. To avoid long user sessions, each task had a maximum time of 2m30s. If the time limit was reached, we considered the attained position and orientation as final and registered it as an unsuccessful attempt. Although some tasks required



(a) Task 1. (b) Task 2.

(c) Task 3. (d) Task 4.

(e) Task 5. (f) Task 6.

Figure 4.26: User evaluation tasks.

only translation or rotation transformations, none of those were restricted on any of these tasks, and the scaling feature of MAiOR was disabled.

The first two tasks required translations only. For the first task, the carbon needed to be moved only along the XX axis. In the second, its position was initially incorrect in all three axes. Third and fourth tasks required rotations only. In the third task participants needed to rotate the carbon about the ZZ axis, while the fourth they needed to perform rotations about XX, YY and ZZ axes. The last two tasks were the most complex, requiring both translations and rotations. The fifth task required the object to be rotated about the ZZ axis and moved along XX and YY axes. On the final task, participants had to apply a full 6-DOF transformation.

Additionally, and solely for MAiOR, we devised a scaling task to gather subjective user feedback. This consisted in changing the scale of three objects, according to the target shown in front of them, as depicted in Figure 4.27. The first required scaling

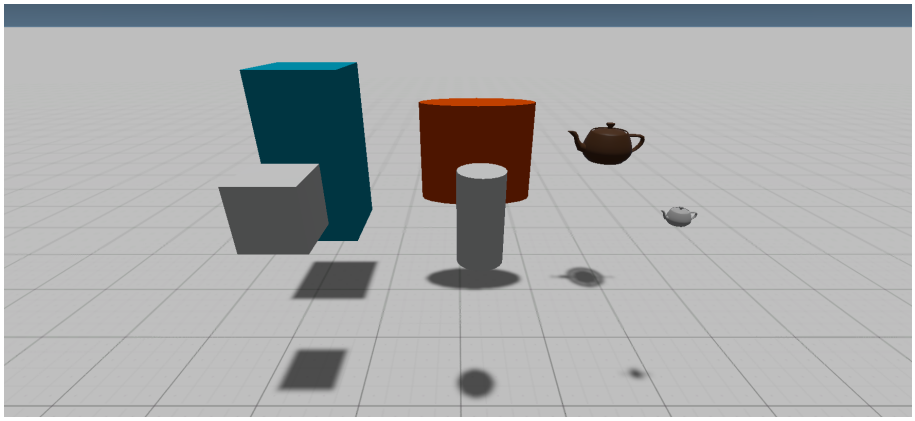


Figure 4.27: Additional task to test MAiOR's scaling transformation.

along the YY axis, the second along the XX axis, and the third along all three axis uniformly. For this task, scaling was the only transformation allowed.

4.3.2.4. Setup and Prototype

In order to compare MAiOR against the two baselines, we developed a prototype where we implemented the three manipulation techniques.

Hardware Setup

We used a setup based on the HTC Vive, which tracks users' head and hands position and orientation in 6-DOF. This setup, although not being totally wireless, offers a tracking accuracy better than that used in previous sections' evaluations. While head tracking is made by the headset itself, hand tracking is made through hand-held controllers. Each controller has a trigger with 10 levels of pressure. We use the trigger to detect if the hand is opened (no pressure), closed (any pressure level from level 1 to 9) or tightly closed (full pressure at level 10). The last pressure level is perceived by a slight click on the trigger, and is used to lock translations and rotations to 1-DOF in MAiOR. This approach to trigger restricted transformations is adaptable to other kinds of force-sensitive devices, such as that used in Sections 4.1 and 4.2, and avoids the use of multiple buttons.

Virtual Environment

We developed the prototype using the Unity3D engine. The environments consists of a wide and empty plane area, with shadows but no gravity nor collisions. Users'

hands are represented through virtual replicas of the controllers. Objects become transparent whenever users intersect them, and became opaque as soon as they are grabbed in order for improved visual feedback.

4.3.2.5. Participants

We had a total of 24 participants, 16 males and 8 female, between the ages of 17 and 33 years old. Most of them hold at least a BSc degree (80%). 83% reported having no previous experience in VR, and 70% had never used any kind of gesture recognition systems, such as Xbox Kinect, Nintendo Wiimote or Playstation Move. Only 16% of the participants use 3D modeling systems at least once a month.

4.3.3. Results and Discussion

During user sessions, we gathered objective data through a logging mechanism and subjective data from the questionnaires. To analyze such data, we used Shapiro-Wilk test to assess data normality. Then, we ran the repeated measures ANOVA test to find significant differences in normal distributed data. Otherwise, we ran Friedman non-parametric test with Wilcoxon Signed-Ranks post-hoc test. In both cases, post-hoc tests used Bonferroni correction (presented sig. values are corrected).

4.3.3.1. Task Performance

We measured success rate, completion time and object placement error (position error in millimeters and orientation error in degrees) for each task.

Success Rate

Success rate was defined by the ratio of participants that were able to place the virtual object below the position and orientation tolerated error and within the time limit. Values are shown in Table 4.3, and statistically significant differences were found (Task 1: $\chi^2(2) = 27.900$, $p < 0.0005$; Task 3: $\chi^2(2) = 13.875$, $p = 0.001$; Task 4: $\chi^2(2) = 8.000$, $p = 0.018$; Task 5: $\chi^2(2) = 8.400$, $p = 0.015$). On task one, SVH had a lower success than MAiOR ($Z = -3.638$, $p < 0.0005$) and Widgets ($Z = -4.243$, $p < 0.0005$). On task three, Widgets outperformed both SVH ($Z = -3.051$, $p = 0.006$) and MAiOR ($Z = -2.887$, $p = 0.012$). On task four, Widgets was

	Task 1*	Task 2	Task 3*	Task 4*	Task 5*	Task 6
MAiOR	79%	58%	33%	42%	17%	13%
SVH	17%	33%	29%	25%	29%	33%
Widgets	92%	54%	75%	67%	54%	17%

Table 4.3: Success rate per task for each technique. * indicates statistical significance.

better than SVH (Z = -2.673, p = 0.024) and, on task five, better than MAiOR (Z = -2.714, p = 0.021).

SVH attained consistent success rate values. Since it does not possess any kind of transformation's restriction, all tasks are of identical execution. The low success rate relates to the difficulty to accurately place an object in mid-air using an exact mapping. Widgets had the highest overall success rate, successfully preventing undesired transformations. However, and as shown in previous sections, the more complex the task, the higher the time needed to complete it, and more difficult it is to do so within the time limit. MAiOR's success rate also decreased when task complexity increased. We believe that this is due to both DOF separation's additional steps, and a not so simple rotation metaphor that made all tasks involving rotations more difficult. Indeed, the success rate increased from task 3 to task 4, suggesting that experience may have affected participants' performance in rotation tasks.

Completion Time

To analyze completion time, we only considered times attained by participants who achieved tasks' goal within the time limit, which are depicted in Figure 4.28. Although we did not find any statistical significant differences, we verified in tasks 2 and 4 a tendency for MAiOR (Task 2: avg=48s; Task 4: avg=39s) to be faster than Widgets (Task 2: avg=92s; Task 4: avg=72s). Taking into account individual times, 8 out of 9 (task 2) and 6 out of 6 (task 4) participants that completed those tasks with both techniques had lower completion times with MAiOR than with Widgets. However, this needs to be confirmed with further testing.

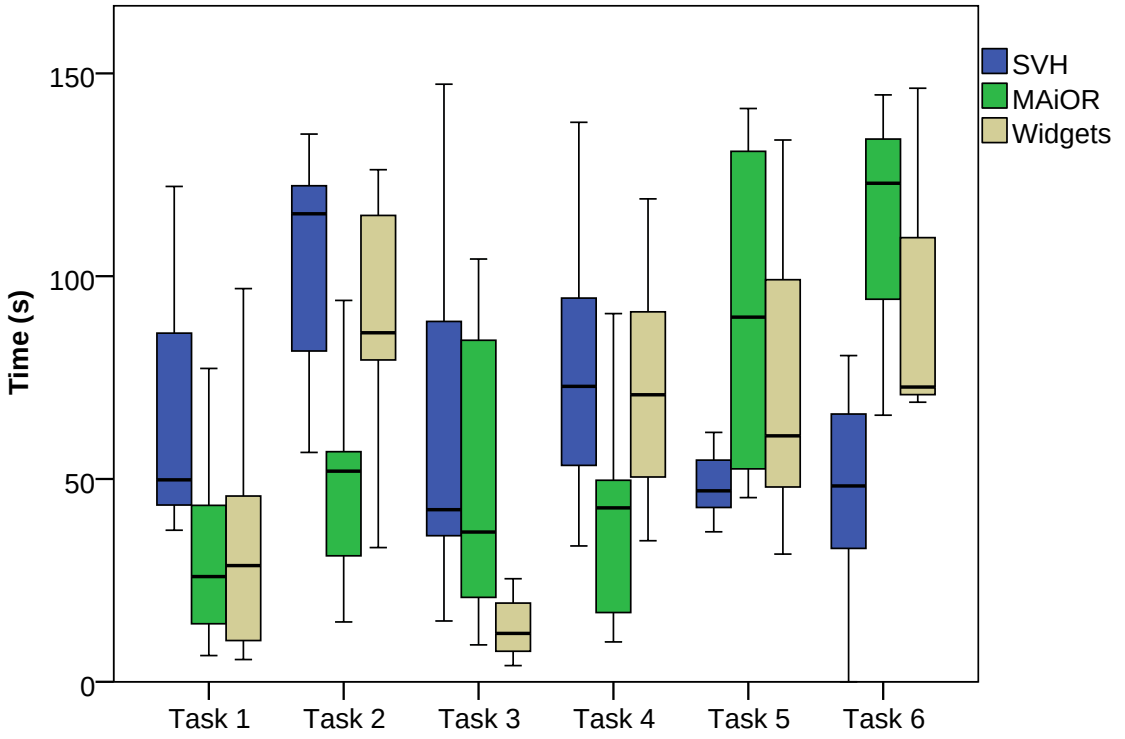


Figure 4.28: Tasks' completion time, in seconds. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

Placement Error

Regarding placement error, we analyzed both position and rotation error (Figures 4.29 and 4.30). For participants that did not achieve tasks' goal, we considered the current placement when the time expired. Significant differences existed for position error (Task 1: $\chi^2(2) = 29.84$, $p < 0.0005$; Task 3: $\chi^2(2) = 20.118$, $p < 0.0005$; Task 4: $\chi^2(2) = 19.5$, $p < 0.0005$; Task 5: $F(1.363, 19.085) = 10.075$, $p = 0.003$; Task 6: $\chi^2(2) = 9.264$, $p = 0.01$). Post-hoc tests showed that for the first task, SVH (average: 4.65mm) was less accurate than both Widgets (average: 0.43mm, $Z = -4.108$, $p < 0.0005$) and MAiOR (average: 0.55mm, $Z = -3.921$, $p < 0.0005$). In task 3, SVH (average: 3.44mm) performed worst than Widgets (average: 0.43mm, $Z = -4.108$, $p < 0.0005$). In both fourth and fifth tasks, Widgets (Task 4 average: 0.00mm; Task 5 average: 0.69mm) outperformed SVH (Task 4 average: 3.38mm, $Z = -3.725$, $p < 0.0005$; Task 5 average: 2.61mm, $p = 0.001$) and MAiOR (Task 4 average: 2.94mm, $Z = -2.803$, $p = 0.015$; Task 5 average: 5.00mm, $p = 0.006$). However on the sixth task, SVH (avg=2.61mm) proven to be more precise than Widgets (avg=6.52mm, Z=-3.015, p=.009) and MAiOR (avg=4.41mm, Z=-2.521, p=.036), mostly due to time constraints. In rotation only tasks (task 3 and 4), the existence of a positional error with MAiOR implies that participants performed translations,

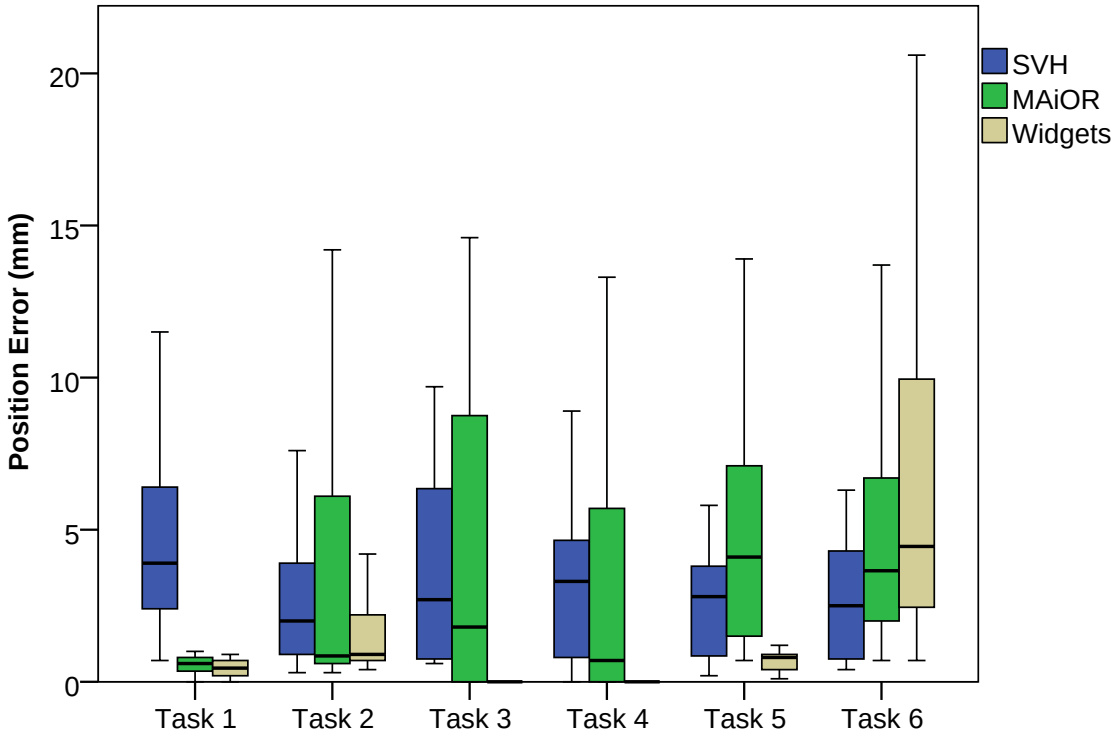


Figure 4.29: Position error, in millimeters. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

even though MAiOR had transformation separation. This is further discussed in Section 4.3.3.3.

Technique used also influenced orientation error (Task 1: $\chi^2(2) = 32$, $p < 0.0005$; Task 2: $\chi^2(2) = 22$, $p < 0.0005$; Task 3: $\chi^2(2) = 13.857$, $p = 0.001$; Task 4: $\chi^2(2) = 8.4$, $p = 0.015$; Task 5: $\chi^2(2) = 10$, $p = 0.007$). In translation only tasks, SVH (Task 1 average: 1.77°; Task 2 average: 1.22°) was the only that caused this kind of error, significantly worse than Widgets (Task 1 average: 0.00°, Z = -3.92, $p < 0.0005$; Task 2 average: 0.00°, Z = -3.408, $p = 0.003$) and MAiOR (Task 1 average: 0.00°, Z = -3.516, $p < 0.0005$; Task 2 average: 0.00°, Z = -3.517, $p < 0.0005$), which shows the benefits of translation and rotation separation in preventing unwanted transformations. In the remaining three tasks Widgets (Task 3 average: 0.54°; Task 4 average: 0.77°; Task 5 average: 0.48°) achieved better results than SVH (Task 3 average: 2.42°, Z = -3.743, $p < 0.0005$; Task 4 average: 1.39°, Z = -2.722, $p = 0.018$; Task 5 average: 1.47°, Z = -3.637, $p < 0.0005$), as expected. In these tasks, Widgets was also more precise than MAiOR (Task 3 average: 1.72°, Z = -2.92, $p = 0.012$; Task 4 average: 1.27°, Z = -2.442, $p = 0.045$; Task 5 average: 2.20°, Z = -2.442, $p = 0.045$), possibly due to the difficulty reported by participants to perform rotations with this technique.

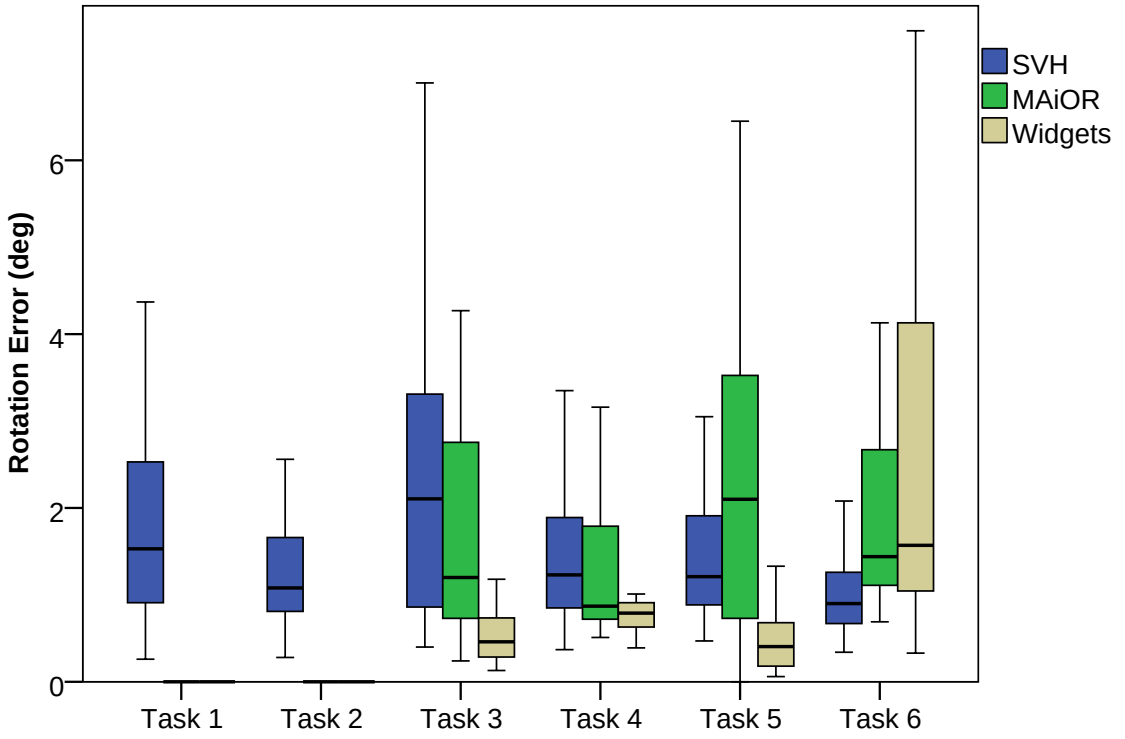


Figure 4.30: Rotation error, in degrees. The chart present the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

4.3.3.2. User Preferences

Through questionnaires, we asked participants about their experience with each technique. This included general ease of use, fun factor, ease of manipulating object position and orientation, recall regarding translation and rotation transformations and overall fatigue. In all questions it was used a Likert Scale from 1 to 5, being 5 the favorable value. Answer's are reported in Table 4.4.

We found statically significant differences in ease of use ($\chi^2(2) = 6.685$, $p = 0.035$), translation recall ($\chi^2(2) = 11.541$, $p = 0.003$), rotation easiness ($\chi^2(2) = 14.427$, $p = 0.001$) and recall ($\chi^2(2) = 20.738$, $p < 0.0005$), and fun factor ($\chi^2(2) = 7.69$, $p = 0.021$). Participants agreed that Widgets were generally easy and more fun to use than MAiOR (easiness: $Z = -2.623$, $p = 0.027$; fun: $Z = -2.599$, $p = 0.009$). While for translations there was no significant difference in easiness between the three techniques, participants agreed that remembering how to translate was easier with Widgets than with MAiOR ($Z = -3.246$, $p = 0.003$). Participants also indicated that it was harder to perform and to remember rotations with MAiOR than both SVH (rotation easiness: $Z = -3.407$, $p = 0.003$; rotation recall: $Z = -3.578$, $p < 0.0005$) and Widgets (rotation easiness $Z = -2.99$, $p = 0.009$; rotation recall: $Z = -3.691$,

	MAiOR	SVH	Widgets
Overall easiness *	2 (1)	2 (1)	3 (2)
Translation easiness	3.5 (2)	4 (2)	4 (2)
Translation recall *	3.5 (1)	5 (1)	5 (1)
Rotation easiness *	2.5 (1)	3.5 (2)	3 (2)
Rotation recall *	3 (1)	5 (1)	4,5 (1)
Fatigue	3 (2)	3 (2)	3 (1)
Fun *	3 (1)	4 (1)	4 (2)

Table 4.4: Answers to questionnaires, regarding each criteria (median, interquartile range). * indicates statistical significance.

$p < 0.0005$). Differences in recall can be justified by the complexity inherent to the gesture-based grammar, as the available actions are not visible to users and an additional effort must be done to remember how to perform such actions. Contrary to MAiOR, with Widgets and SVH users only needed to remember to grab and drag the object. Moreover, the rotation metaphor adopted in MAiOR was identified as being difficult to get acquainted. According to participants' comments, they needed more time to fully take advantage of MAiOR.

Regarding MAiOR's scaling feature, we also asked participants to answer a questionnaire about its overall easiness, fun factor, ease of recalling its operation, and fatigue caused. Answers' are shown in Table 4.5, where a higher value on the Liker scale indicates a more favorable opinion. Generally, participants had a positive judgment on performing scaling transformations with MAiOR. They also stated that it was a

	Classification
Overall easiness	4 (1)
Recall	4 (1)
Fatigue	5 (1)
Fun	4.5 (1)

Table 4.5: Participants' classification of MAiOR's scaling transformation (median, interquartile range).

feature easy to understand because it works like typical scaling or zoom operations found in today's smartphone and tablet applications.

4.3.3.3. Observations

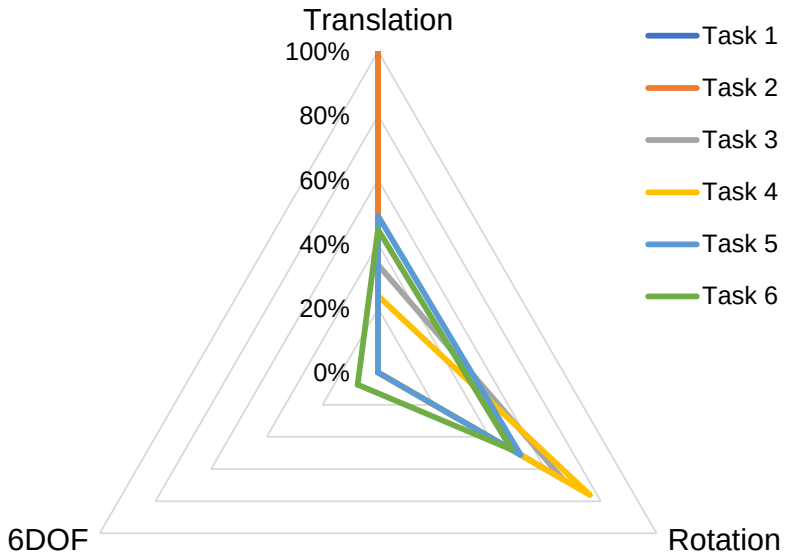
We observed that, when using the direct SVH approach, some participants used specific poses to help reducing hand tremor and to be easier to achieve tasks' goal. They held tight the dominant hand's arm while using the other arm as support, as shown in Figure 4.31. Other participants mostly resorted to chance. First, they roughly placed the object with position and orientation close to the target, then tried successive grabs and releases hoping one would be within the acceptable threshold.

As previously stated, in tasks 2 and 4 almost all participants had better performance with MAiOR than Widgets. Looking at participants' profiles, we found that in task 2 they had several distinct backgrounds. This suggests that MAiOR's translation approach might be adequate for novice users. On task 4, however, they had backgrounds related to 3D modelling, such as design and architecture. This can mean that MAiOR's rotations require more experience with 3D manipulation concepts, such as rotation axes. More, we noticed that participants had difficulties understanding MAiOR's circumference widget's motion. This might be due to the fact

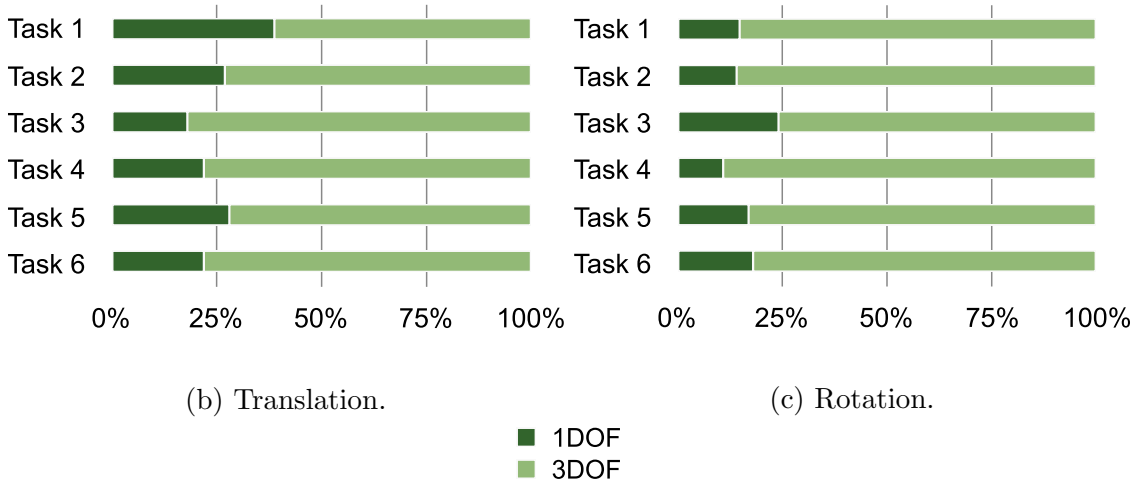
Figure 4.31: Example of a pose performed by participants.

that wrist rotation was not accounted for this. One of the most experienced participants also reported that using a custom pivot for rotations instead of the object's center could be beneficial.

Additionally to the task performance measures presented in Section 4.3.3.1, we compared completion time and position error between participants that used the scaled translation mapping to achieve target's position with MAiOR and those who did not, for the translation only tasks (tasks 1 and 2). We found no statistically significant differences, which suggests that, when a highly accurate tracking solution is available, scaling users' movements does not provide better precision.



(a) Transformations.



(b) Translation.

(c) Rotation.

Figure 4.32: Time distribution, in percentage, between transformations (a), and between 3-DOF and 1-DOF translation (b) and rotation (c), in MAiOR.

Besides completion time, we logged where participants spent time when manipulating the object for all tasks. This is illustrated in Figure 4.32. It is possible to see that in rotation only tasks (task 3 and 4), participants' first instinct was to directly grab the object. This originated some position error that they tried to correct. Also, users favored transformation separation, which is mostly noticed in tasks 5 and 6, where time spent performing independent translations and rotations is evenly distributed. Simultaneously 6-DOF transformations were barely used.

We also registered, for both independent translations and rotations, time spent in 3-DOF and 1-DOF manipulations in MAiOR, also depicted in Figure 4.32. We verified that participants preferred to perform transformations in 3-DOF, as they were faster than 1-DOF. These results contradict previous research suggestions [125]. With MAiOR's scaled movements in translation, participants found the positioning precision attained with 3-DOF adequate. On the other hand, they found it difficult to achieve the required orientation, yet they rarely restricted rotation to a single DOF. Additionally, the tight grab gesture used to constraint transformations originated unintentional activations of 1-DOF, as sometimes participants applied the extra strength without noticing.

4.3.4. Lessons Learned

In this section, our objective was to assess if personalized transformation axes can contribute for fast yet accurate manipulations. For this, we developed a novel technique, MAiOR, and compared it against two baseline approaches, one based on direct 6-DOF manipulations and other on 3D widgets. While MAiOR did not compromised tasks' completion time, Widgets had the best performance overall. Even though, MAiOR showed promising results on isolated transformation tasks.

When analyzing individual transformation times in MAiOR, we found that while participants took advantage of transformation separation to prevent unwanted results, they did not sought single DOF manipulations. Moreover, although MAiOR reduced visual clutter over the widgets-based approach, its interaction grammar was harder to recall. We reckon that, with a longer learning period, MAiOR's task performance can increase as users grasp on how to take advantage of everything MAiOR can offer.

With this study, we can conclude that, if a reliable and accurate spatial tracking solution is used, the most important consideration for precise mid-air manipulations is the support for transformation separation. For object translation, single DOF

control is only advantageous for tasks where the required displacement is coincident to one of the axes from the object's reference frame. For more complex movements, 3-DOF translations are preferred and potentially faster. Rotations, on the other hand, require clever approaches for isolated 3-DOF rotations, as this is what appears to be more difficult to perform. Users are not accustomed to rotate objects without translating them also, in order to fix an incorrect positioning that only became clear after rotating. A possibility is to explore pivot points for rotation.

With the attained results, scaled mappings and 1-DOF manipulations seem to be useful just to prevent jitter from less reliable trackers. These findings, in conjunction with those from the previous section, disprove our second and third hypotheses.

4.4. Chapter Summary

Direct approaches for mid-air object manipulation, albeit natural and fast, lack precision. In this chapter, we studied ways to increase the precision of mid-air manipulations in tasks that require translations and rotations. Firstly, we compared an approach for single DOF control using 3D virtual widgets, similar to those used in traditional mouse-based interfaces, against an approach with an exact mapping and another with a scaled mapping. For this, we developed a fully-immersive prototype resorting to depth cameras for user tracking. Through a user evaluation, we saw that scaled mappings were only suited for translations, as they made rotations more difficult to grasp. Widgets did increase precision over the others, but also increased task completion time in relation to the direct approach.

From these results, we developed a novel technique, WISDOM, combining the best characteristics from each evaluated technique. It allows users to toggle widgets on and off. When disabled, 6-DOF manipulations can be performed directly on the object. When enabled, users can execute 3-DOF and 1-DOF translations, both dynamically scaled, and single DOF rotations. However, due to its lengthy list of features, participants found it hard to recall all of WISDOM's possibilities, and it did not perform better than the widgets-based approach.

Then, we explored custom transformation axes, in an attempt to reduce task completion time while maintaining the precision offered by single DOF manipulations. We conceived MAiOR, a technique that decides which transformation is going to be applied depending on whether the interaction started inside or outside the object,

for translation and rotation respectively. These transformations begin to be applied in 3-DOF, but can be restricted to 1-DOF at users' will, and the transformation axis is defined by the movement performed until then. Translations can have a scaled mapping or be turned into 6-DOF through explicit user input. In this evaluation, we resorted to a highly accurate user tracking solution. Results showed that 1-DOF manipulations and scaled translations did not contribute for improved performance in such settings, and that separating translation and rotation are key in preventing unwanted transformations that originate placement error.

5

Out-of-Reach Interaction

Interactions with virtual objects in IVEs are usually done by directly reaching them with the hand. Naturally, objects that are outside arms-reach pose an additional challenge. Being capable of selecting such objects is the first requirement to be able to manipulate them, and not always navigating to near them is possible or appropriate. Common approaches to perform out-of-reach selections require users to point at the objects using ray-casting, or follow an arm-extension metaphor. However, these approaches can be severely impacted by tracker jitter and hand tremor.

In this chapter, we focus on the challenge of effectively selecting out-of-reach objects in IVEs. We believe that an iterative progressive refinement strategy can be successfully implemented in such environments. For this purpose, we developed PRECIOUS, a novel out-of-reach selection technique for IVEs. In the next section, we present PRECIOUS, as well as its validation through a user evaluation against existing techniques.

5.1. Employing Iterative Refinement in IVEs

To reduce the impact of using inaccurate input, selection volumes such as a cone [74] or a pyramid [84] can be used instead of a ray to point at distant objects. Nevertheless, this still has drawbacks: if the volume's size is too small it will continue to suffer from jitters and tremors, and if it is too large several objects will be intersected, which requires some sort of approach for disambiguation.

Selection techniques that improve upon ray or volume casting have been proposed to interact with large scale displays. For instance, with iterative progressive refinement users can select a group of objects, which are then rearranged into smaller groups on the screen [26, 70]. The user can repeatedly select one of the smaller groups until there is only one object left. As an alternative, zoom techniques can make objects appear closer, easing selection tasks [9]. However, these techniques cannot be naively used in immersive virtual environments (IVEs): rearranging objects might disrupt immersion, and reducing the field-of-view can cause discomfort [73].

In this section, we propose PRECIOUS, a novel approach that offers iterative progressive refinement in IVEs as a way for accurate and time consistent selections of far placed objects. We allow the user to select groups of objects using a cone-casting approach, then we instantaneously move the user closer to the objects. This process can be iteratively repeated, until the user can select the desired object at ease. We begin by describing PRECIOUS. We follow with a user evaluation conducted to validate PRECIOUS, where we compared it against two techniques from literature, which follow ray-casting and arm extension metaphors. Lastly, we present and discuss the results from this evaluation and the lessons learned.

5.1.1. Proposed Technique: PRECIOUS

In order to support an iterative progressive refinement strategy in IVE for out-of-reach object selection, we developed PRECIOUS (Progressive REfinement using Cone-casting in Immersive virtual environments for OUt-of-reach object Selection), exemplified in Figure 5.1. It offers an infinite reach, using an egocentric virtual pointer metaphor [104]. We use a cone as a selection volume, casted from users' hand. While pointing, users can make the cone aperture wider or smaller, and change the cone's reach. Objects that fall inside the cone will be selected. Users are then moved closer to the selected objects for a more accurate selection. As such, this can

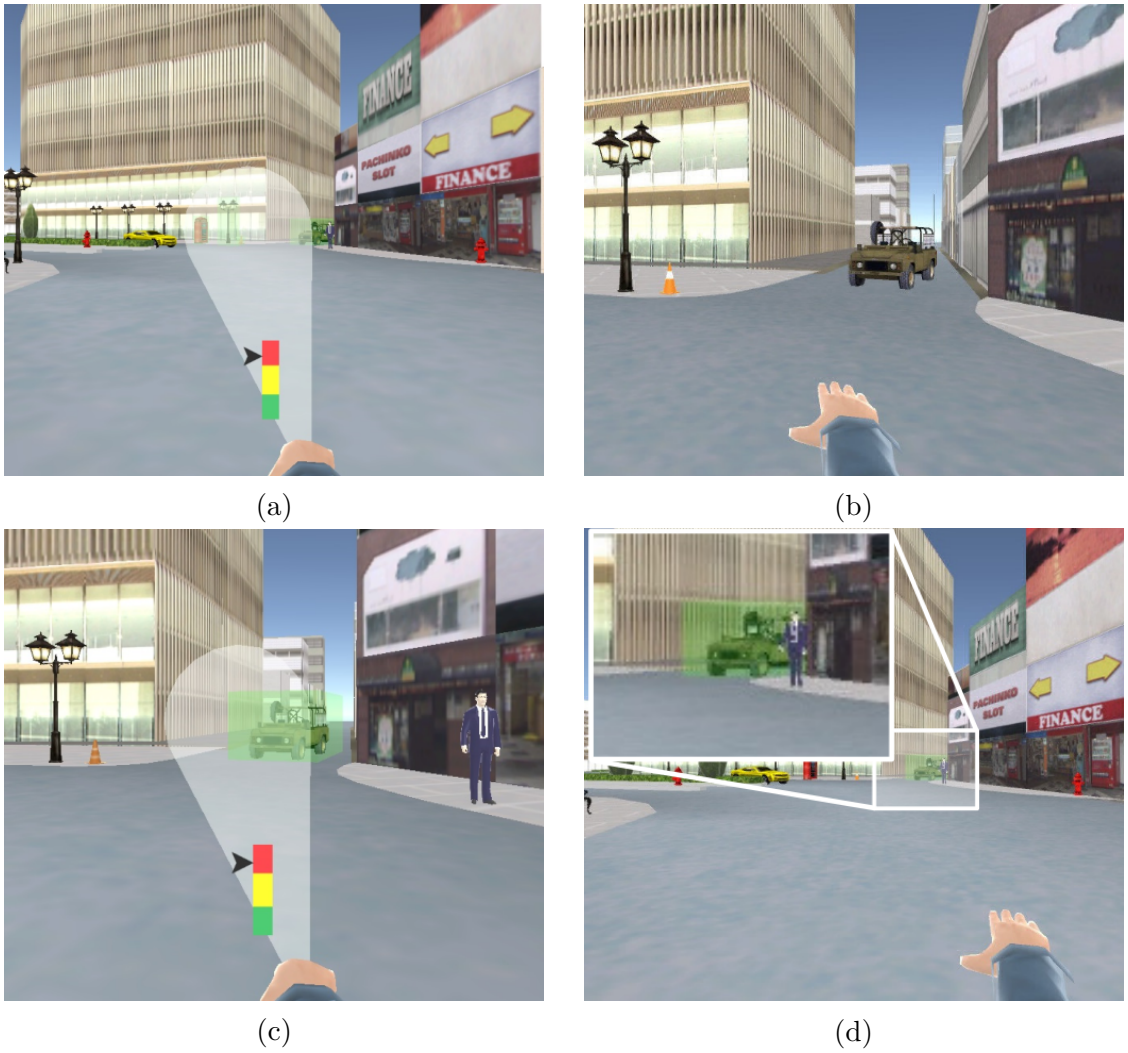


Figure 5.1: PRECIOUS technique: selection cone intersecting various objects (a), refinement phase, moving the user closer to the objects (b), single object selection (c), returning to the original position with the object selected (d).

be considered as a possible immersive implementation of Discrete Zoom [9], although we modify users' position instead of the field-of-view. This process is repeated until a single object is selected or, if users desire, can be stopped at any time to select a group of objects, supporting both single and multiple cardinality.

To help users better understand which objects are inside the cone volume, our approach highlights them showing their bounding boxes as semi-transparent green cubes. Next, we detail how the selection process can be performed with PRECIOUS. We first describe how the cone can be manipulated, then we specify how the progressive refinement works, and how users can select multiple objects simultaneously.

5.1.1.1. Selection Volume Manipulation

To define the selection volume, we resorted to a flashlight metaphor [74]. We cast a cone from users' dominant hand, which is used as a selection volume. The orientation of the hand defines the direction of the cone. We also offer two modifications users can perform on the cone: the first is to control its aperture and the second is to change its reach.

Aperture

While the Aperture Selection technique [41] allows users to change the cone's aperture by moving their hand backwards and forwards, we use instead the rotation of the users wrist (Figure 5.2). This is because the origin of PRECIOUS' cone is placed in users hand and not in users' eye point. The initial aperture is 11 degrees. When the wrist is rotated clockwise, the aperture of the cone increases until the opening angle reaches 15 degrees. Analogously, if rotated in the opposite direction, the aperture will decrease until a 7 degrees angle is achieved.

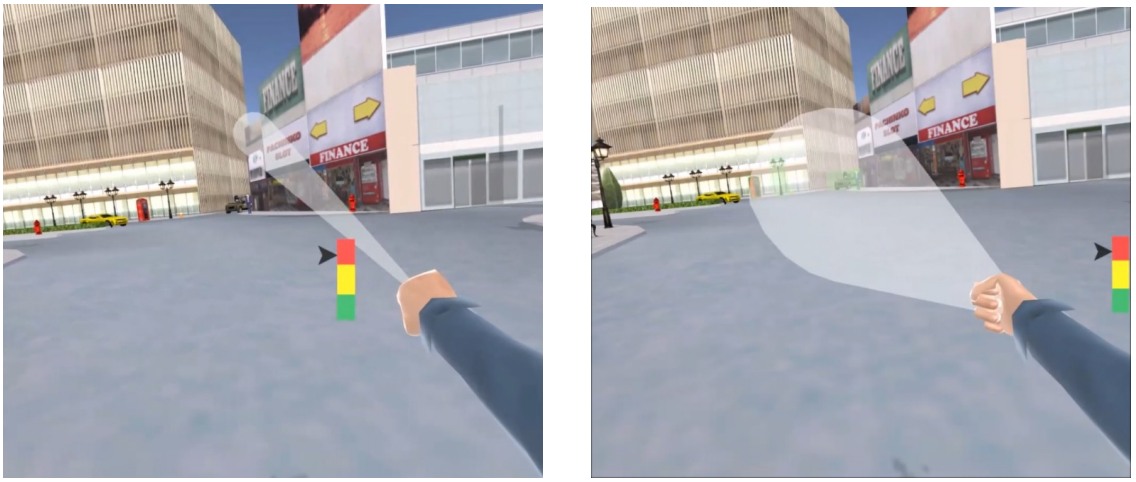


Figure 5.2: Controlling the aperture of the cone.

Reach

To manipulate the cone's reach, we adopted a similar approach to the one used on Stretch Go-Go [20] to control users' virtual hand. As such, we define three spherical regions around the user (Figure 5.3), but we center them in the hip side corresponding to the dominant hand. When users extend their hand into the outermost region (more than 50 cm from the shoulder), the cone will stretch in the pointed direction

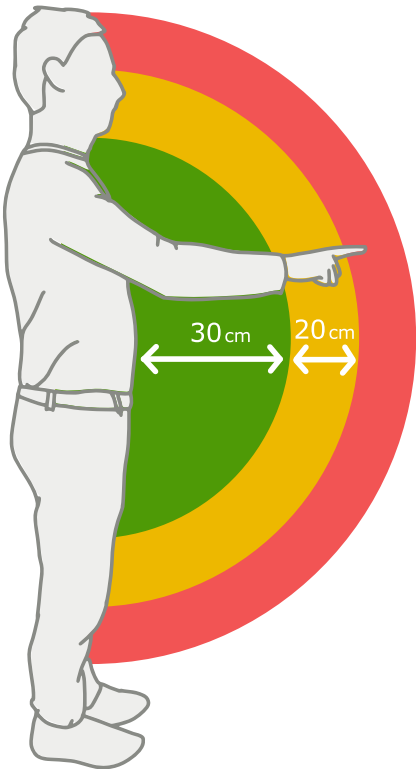


Figure 5.3: Distances regions for cone's reach control.

at a rate of 5 m/s. Placing the hand in the innermost region (less than 30 cm), will make the cone decrease in size with the same speed. While the hand is placed in the middle region (from 30 to 50 cm), the cone's reach remains unchanged.

To help users understand in which region their hand current is, we show a widget when the cone is active. The widget shows the three regions with an arrow pointing towards the one currently active, also depicted in Figure 5.2. Differently from Stretch Go-Go, we use a diegetic UI showing this widget near the users' hand. This way the widget is always visible when users are controlling the cone.

5.1.1.2. Progressive Refinement

The usage of a selection volume instead of a ray can lead to several objects being intersected by it. When this happens, a disambiguation mechanism is triggered. To give users total control over the selection, we follow an iterative progressive refinement approach. In our approach, we drew inspiration from previous zoom techniques [9], but instead of changing the camera's field-of-view, we move users closer to selected objects in the virtual world.

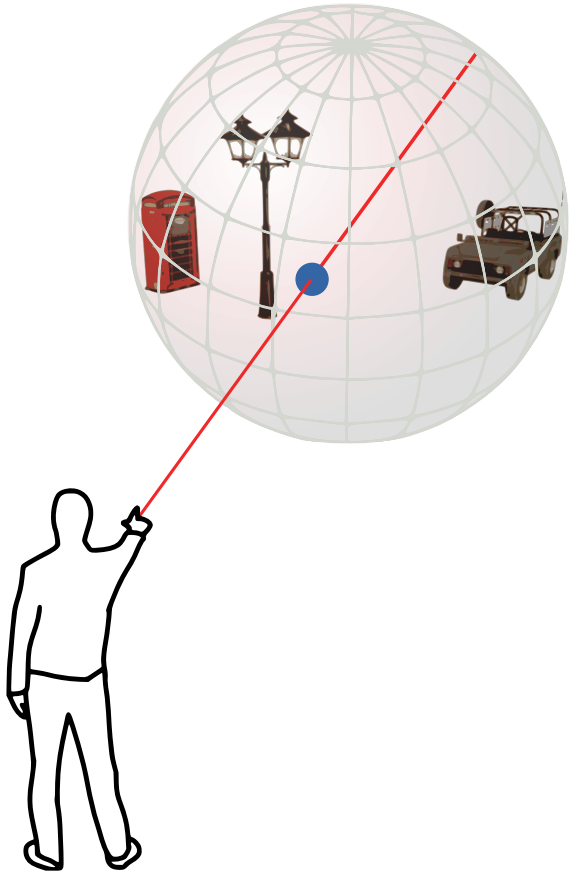


Figure 5.4: Refinement process: the blue dot represents where the ray intersects the sphere, and defines next user position.

To calculate the new users' position (Figure 5.4), we surround all selected objects with a sphere. This sphere is then intersected with a ray casted from users' hand similarly to the cone. The intersection point gives the position where users will be moved to. To move users we perform an instantaneous teleport action (also known as infinite velocity), which showed better results regarding user disorientation and discomfort than other animated techniques, as described in Appendix B. The process is repeated until two or less objects are selected.

When two objects are very close to each other, it might be difficult to manipulate the selection cone in such a way that it only intersects a single object. To prevent user frustration we made these final stages of the refinement process easier. Following a canvas disambiguation [37] approach, we place them side-by-side in front of the user, while hiding the remaining objects in the scene (Figure 5.5).

The object that is closer to the user is displayed first, on the left. Although an higher number of objects could be used to trigger this final step, we opted to perform it

Figure 5.5: Double Selection process.

only when the cone intersects two, so that user immersion is disrupted as little as possible.

After a single object is selected, the refinement process is over and users are placed back in their starting position. The object remains selected and it is ready to have further actions applied to it.

5.1.1.3. Multiple Object Selection

Although initially conceived for single object selection, PRECIOUS also allows multiple object selection. While pointing with the cone, several objects might be intersected. In such cases, users can either start the refinement process described above, or they can select all objects at once. In the latter case, the refinement process is promptly concluded, users return to their original position and objects are kept selected.

5.1.2. User Evaluation

To validate PRECIOUS, we compared it against two techniques from literature: Stretch Go-Go [103], which follows an arm-extension metaphor, and Flashlight [74], that uses cone-casting and an heuristic disambiguation method for single step progressive refinement. The choice of these techniques is explained because of their out-of-reach selection and infinite selection capability.

While a panoply of other techniques exist, they are not suitable for our scenario. For instance, Aperture gives different weights to the intersected objects [41], which is not always appropriate, while others require additional interactive surfaces on the disambiguation phase [86]. Others such as the SQUAD [70] VR implementation and Disambiguation Canvas [37], completely change the virtual environment, which may disrupt immersion. Techniques that perform continuous progressive refinement also present additional problems. For example: Shadow-cone [119] require users to point at the desired object from different positions, but this is difficult to do when the object is very far; Zoom techniques [9], although having been developed for non-immersive and non-stereo environments, could be implemented in IVE, but changing the field-of-view might lead to user discomfort or cybersickness.

5.1.2.1. Baseline Techniques

While our technique implements an iterative progressive refinement in IVE, built upon Discrete Zoom [9], both Stretch Go-Go and Flashlight, the chosen baseline techniques, follow different approaches. Stretch Go-Go was one of the first techniques developed to overcome the physical limitations of out-of-reach selection. It uses the metaphor of extending users' arm [103], with an infinite reach. While pointing, a virtual hand is continuously moved outwards when users extend their arm, or inwards when users retract it. A gauge is shown indicating the current action being applied to the virtual hand according to users' physical hand: red when moving away from users, yellow when the distance is kept unchanged, and green when getting closer to users. When objects are intersected by the virtual hand, they can be then selected.

The Flashlight technique was developed with the intention to overcome the low accuracy of ray-casting [74]. As the name suggests, it uses a flashlight metaphor, using a cone as selection volume instead of a ray. Objects that fall inside the cone are candidates for selection. Keeping its roots in ray-casting, when more than one object are hit, the closest to a ray in the center of the cone is selected. If two or more objects have the same distance to the center ray, the object with the smaller Euclidean distance to the user is chosen. We only highlight the object that will be selected at that time. We did not allow any modifications to the cone, as previously proposed [41], since it would require additional disambiguation mechanisms, and we want users to explicitly define which object they want to select. Cone's aperture was set to 7 degrees.

5.1.2.2. Procedure

All user evaluation sessions followed the same methodology and lasted approximately 45 minutes. The experiment was carried out in our laboratory, a controlled environment. We started with a brief introduction, and then, for each technique, we explained them and played a video illustrating how they work. Participants were then given a training period of three minutes to adjust themselves to the environment and to the technique about to be tested. Afterwards, participants were instructed to perform four tasks, described in the next section. After completing these tasks for each technique, participants were asked to fill out a questionnaire about their experience. Techniques followed an alternated order, using a Latin square design so that all permutations were exhausted, to avoid biased results. In the end, participants filled out a profiling questionnaire.

5.1.2.3. Tasks

Participants were requested to complete a set of four tasks for each technique, and all consisted in selecting a cactus in our virtual environment (Figure 5.6). In these tasks, we had two variables for target object positioning: distance from the user and amount of surrounding objects. For tasks where the cactus was closer to the user we used distances that can be considered to be plausible in room-sized scenarios, whereas in the tasks with the object far from the user we placed it on the other side of a large avenue. Although not exploring heavily cluttered environments as other works [9, 37, 70], where a lot of objects are placed together in a small space, half of our tasks in included some distractors. This way, we explored both situations where cone-casting approaches could easily select the target object alone, and others where this would be considerably more difficult, requiring refinement strategies. Tasks were designed to have an increasing difficulty.

In the first task, the cactus was next to the user, with few objects surrounding it at a considerable distance. For the second, the cactus was placed far from the user, also with few objects next to it. The third task brought the cactus back to the user once again, but increased the number of surrounding objects. Finally, the fourth task, the most difficult, the cactus was placed far from the user with other objects positioned very close to it.

Every time participants selected an object that was not the cactus, we registered it as an incorrect selection. In order to avoid an excessive session duration, we

(a) Task 1

(b) Task 2

(c) Task 3

(d) Task 4

Figure 5.6: Tasks performed by the participants. The square indicates the target cactus.

restricted the duration of each task to three minutes. If participants reached the time limit they would be informed they could stop, and we registered this as an uncompleted task.

5.1.2.4. Setup and Prototype

To validate the adequacy of PRECIOUS for selecting out-of-reach objects in immersive virtual environments, we developed a prototype.

Hardware Setup

The setup used for our prototype is composed of several components. To gather user input, we used non-invasive and affordable tracking hardware. It tracks users' full body using three Microsoft Kinect V2 depth cameras and our custom tracker, described in Appendix A. We also apply a double exponential smoothing filter [8] to reduce noise effects from received data.

For increased hand tracking, we used the same custom Arduino based device from the evaluations conducted in Sections 4.1 and 4.2, which includes an IMU and Bluetooth LE modules. The device is placed in the users hand with an acrylic clip, that assures it does not fall when the hand is opened. It tracks 3 DOF orientation and features a pressure pad, which is used to detect if the hand is opened or closed. We start an object selection action when we detect pressure being applied. This allows users to use a natural pointing gesture, as depicted in Figure 5.7. The pressure pad is also able to clearly distinguish two pressure levels. Using these pressure levels, we require users to close their hand with added pressure to trigger multiple object selection.

For the visualization component, we used a Samsung Gear VR headset with a Samsung Galaxy S6 smartphone. This headset tracks head orientation with 3 DOF. This data is combined with depth cameras' information to fully track users point of view with 6 DOF. Communication between tracking hardware and the headset is done using dedicated wireless connection.

Figure 5.7: Pointing with our custom device.

Virtual Environment

We developed our prototype using the Unity 3D game engine. To explore out-of-reach selections with multiple distances, we chose to overcome the size limitation of the test room by using a virtual representation of an urban environment. For this, we resorted to a replica of the Osaka city, in Japan. We took inspiration from urban planning tasks. This urban scenario in our virtual environment has several objects placed in a familiar fashion and, except buildings and pavement, all objects comprised in the environment were selectable. For user representation, we used a full-body avatar, which is totally animated according to tracking information.

5.1.2.5. Participants

We counted with a total of 18 participants (two female), with ages varying between 18 and 40 years old, with the majority (62%) being between 18 and 25. More than half held at least a Bachelor degree (62%). When asked regarding previous experience in Virtual Reality, 39% reported having none. Only 28% admitted never interacted before with a mid-air gesture-based system, such as the Microsoft Kinect or the Wii Remote.

5.1.3. Results and Discussion

During the experiment we gathered user performance data, using system logs for each task, and user preference data through questionnaires. Logs registered information regarding the completion time and the number of incorrect selections. Additionally, we calculated techniques' success rate for each task. Using the Shapiro-Wilk test, we assessed the normality of the data. A repeated measures ANOVA test was then carried out to find significant differences in normal distributed data. Additionally, for data without such distribution, we used the Friedman non-parametric test with Wilcoxon Signed-Ranks post-hoc tests. Both with ANOVA and Friedman tests, post-hoc tests used the Bonferroni correction (presented sig. values are corrected).

5.1.3.1. Task Performance

We measured the total time that participants took on each task, as well as the number of incorrect selections made. Time was registered in seconds and is depicted in Figure 5.8. We also registered the number of incorrect selections, presented in Table 5.1. The success rate of the techniques was also analyzed.

We found statistical significance in the completion time of all tasks (Task 1: $\chi^2(2) = 17.375$, $p < 0.0005$; Task 2: $F(1.013, 8.102) = 18.327$, $p = 0.003$; Task 3: $\chi^2(2) = 19$, $p < 0.0005$; Task 4: $t(9) = -3.802$, $p = 0.004$). We used a Paired T-Test for the fourth task times because there was not sufficient data from Stretch Go-Go to perform a Friedman test (only four participants finished the task, being a sample too small to be tested).

When comparing the completion times in the first task, post-hoc test showed that the Flashlight approach (average: 10s) was faster than both PRECIOUS (average: 22s, $Z = -2.430$, $p = 0.045$) and Stretch Go-Go (average: 44s, $Z = -3.479$, $p = 0.003$), and PRECIOUS to be faster than Stretch Go-Go ($Z = -3.574$, $p < 0.0005$). In this task, all techniques achieved a 100% success rate.

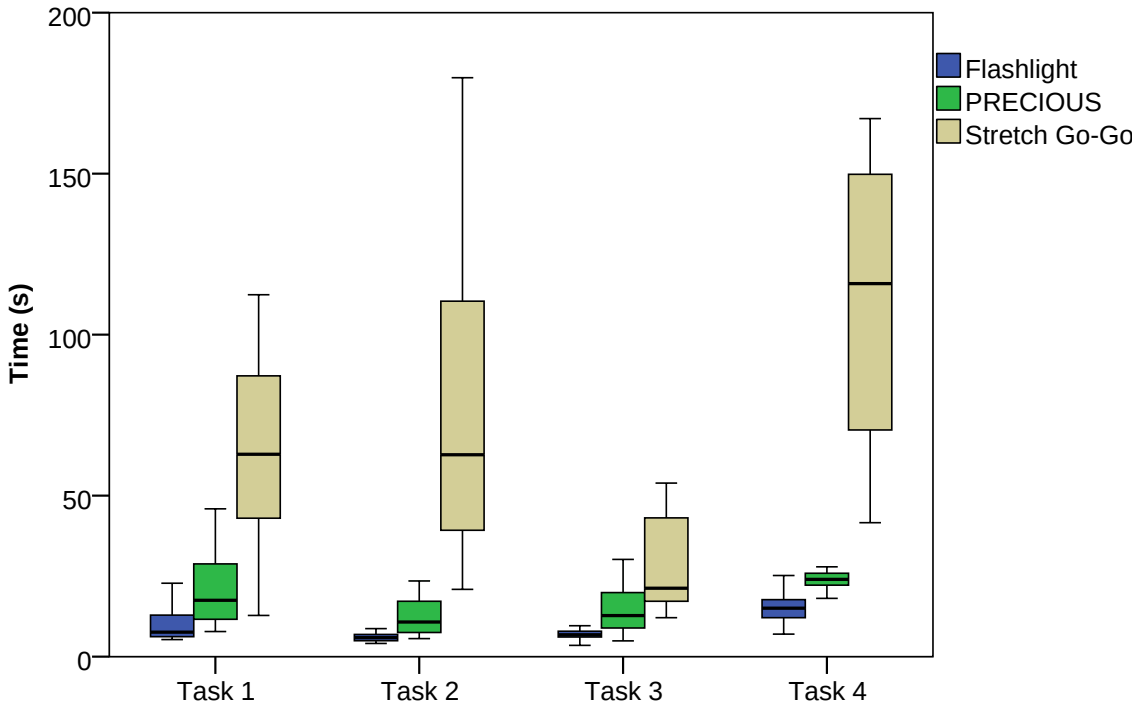


Figure 5.8: Tasks' completion time. The chart presents the median, first and third interquartile ranges (boxes) and 95% confidence interval (whiskers).

	Task 1	Task 2	Task 3	Task 4
Stretch Go-Go	0 (0)	0 (0)	0 (0)	0 (0)
Flashlight	0 (0)	0 (0)	0 (0)	1.5 (3)
PRECIOUS	0 (0)	0 (0)	0 (0)	0 (1)

Table 5.1: Number of incorrect selections per technique (median, interquartile range).

For the second task, Flashlight (average: 6s) was again faster than Stretch Go-Go (average: 76s, p = 0.007). PRECIOUS (average: 11s) also had significantly better completion times when compared to Stretch Go-Go (p = 0.009). This tasks reveals the flaws associated to the Stretch Go-Go technique, as the object was positioned further away from the user and the success rate dropped to 61%, while others remained with 100%.

In the third task, Flashlight (average: 6s) was faster than PRECIOUS (average: 13s, Z = -3.030, p = 0.006) and Stretch Go-Go (average: 28s, Z = -3.296, p = 0.003). In this task, PRECIOUS also showed better results when compared to Stretch Go-Go (Z = -2.480, p = 0.039). As expected, when the object is moved closer to the user the success rate of Stretch Go-Go increased to 83%, but remaining different from the other techniques' 100%.

In the final task, Stretch Go-Go had a success rate of only 22%, making it a sample too small to be analyzed. The others continued with a perfect success rate score. This task revealed the Flashlight (average: 15s) approach to be once again faster than PRECIOUS (average: 24s). This was the only task in which significant differences in incorrect selections occurred. Flashlight, with which half of the participants committed at least two errors (max: 12 errors), caused significantly more errors than PRECIOUS (Z = -3.21, p = 0.003), where only 6 users made an incorrect selection.

5.1.3.2. User Preferences

Questionnaires asked participants about their experience with each technique. They had questions regarding the difficulty of the techniques, the fun factor, if they felt tired and if there was any discomfort. Additionally they were asked about the control of the cone in the Flashlight and PRECIOUS and the virtual hand in Stretch Go-

	Stretch Go-Go	Flashlight	PRECIOUS
Easiness *	1 (1)	4.5 (1)	4 (1)
Satisfaction *	2 (1)	5 (1)	4 (1)
Physical discomfort *	2.5 (2)	5 (1)	5 (1)
Visual discomfort *	3 (1)	5 (1)	5 (1)

Table 5.2: Participants' preferences (median, interquartile range). * indicates statistical significance.

Go. We used a Likert Scale from to 1 to 5 (5 being the most favourable value), and answers are depicted in Table 5.2.

When analyzing participants' answers, we identified significant differences in ease of use ($\chi^2(2) = 23.524$, $p < 0.0005$), fun factor ($\chi^2(2) = 27.180$, $p < 0.0005$), fatigue ($\chi^2(2) = 18.582$, $p < 0.0005$) and discomfort felt ($\chi^2(2) = 22.189$, $p < 0.0005$). Participants strongly agreed that Stretch Go-Go was the hardest to use (Flashlight: $Z = -3{,}673$, $p < 0.0005$, PRECIOUS: $Z = -3.556$, $p < 0.0005$), less fun (Flashlight: $Z = -3.660$, $p < 0.0005$, PRECIOUS: $Z = -3.572$, $p < 0.0005$), most tiring (PRECIOUS: $Z = -3.441$, $p = 0.003$) and most discomforting (Flashlight: $Z = -3.342$, $p = 0.003$, PRECIOUS: $Z = -3.475$, $p = 0.003$).

Participants pointed out two characteristics of the technique that contributed to Stretch Go-Go's results: when moving the virtual hand away it becomes smaller until eventually being barely visible; and as it has a small selection volume, accurately placing it so it intersects the desired object can take too long. The final task made this more evident, as the object is placed further away from the user and controlling the virtual hand becomes even more demanding. The impact of both these problems could be reduced by using an increased selection volume, however that would require additional disambiguation mechanisms.

Regarding the difficulty of using the cone in the Flashlight technique, participants responded positively (median: 4, IQR: 1). When questioned about the easiness of controlling the virtual hand, participants' answers justified previous results (median: 1.5, IQR: 2). Regarding PRECIOUS' selection process, participants classified the control of the cone's aperture to be moderately easy (median: 3, IQR: 1), and its reach was classified as being easy to manipulate (median: 4, IQR: 2). However, we observed that most participants did not change cone's aperture, finding the initial aperture adequate for all tasks. Participants also mentioned that they would prefer

cone's reach to begin as far as Flashlight's, which could be reduced only when necessary. The teleport technique used to move users was received very positively (median: 5, IQR: 0).

5.1.4. Lessons Learned

In this study, our objective was to assess if an iterative progressive refinement approach could be used in IVEs, as a mean for effective selections of objects outside users' arms reach. For this purpose, we developed PRECIOUS, a novel technique that uses cone-casting to select groups of objects, and then instantaneously moves the user closer to them, in order to ease selections. We compared PRECIOUS against two baseline techniques from the literature, Stretch Go-Go and Flashlight.

From our results, it is clear that Stretch Go-Go is an ineffective approach when objects are far away from users' reach. Flashlight was revealed as the fastest technique in almost all tasks. Nevertheless, it is more prone to errors when the difficulty of the selection task increases. Depending in the application context, performing unwanted selections can have a severe impact on the outcome, by applying actions to a wrong object. PRECIOUS, on the other hand, offered low error selections with a small increase in task duration. This shows that iterative progressive refinement strategies can be successfully employed within IVEs for out-of-reach selections, and validates our fourth hypothesis.

The starting short reach of PRECIOUS' cone required participants to increase it in all tasks, being the major reason why our technique was slower than Flashlight. We believe that increasing its cone initial reach, and allowing users to reduce it when needing to exclude objects, would make PRECIOUS achieve completion times at par with Flashlight.

5.2. Chapter Summary

The task of selecting an object is present in our everyday life. From desktop interfaces to immersive virtual environments, there is a need to select an object before interacting with it in any way. When considering the current approaches for object selection in IVEs, few tackle the problem of selecting objects at great distances. The most common approaches of ray-casting and arm-extension severely suffer from

jitter problems when the intended object is too far from users' position. Volume selection techniques, on the other hand, can deal more effectively with this problem, but when several objects are close together unwanted selections may occur.

In this chapter, we proposed PRECIOUS, a combination of iterative progressive refinement and cone-casting that allows users to select objects at various distances. We allow selection of more than one object using the cone volume, and then refining the selection with those fewer objects, by teleporting users next to those objects. The cone can also be manipulated. Besides changing its origin and direction, users can change its reach and aperture. The refinement can be repeated until a single object is easily selected or, if users intend to, interrupted to select a group of objects.

A formal user evaluation was then conducted, where PRECIOUS was compared with two other out-of-reach selection techniques. With the results from this evaluation we found that arm-extension approaches are impractical when selecting objects that are very distant. Simple volume selection techniques can provide faster completion times on standard selection tasks, but when there are objects close to the desired one they are prone to incorrect selections. Regarding PRECIOUS, we can state that although it was not the fastest, the lack of errors and uniform completion times across all scenarios tested make it a suitable out-of-reach selection technique.

6 Conclusions and Future Work

Object manipulation is one of the most relevant tasks in virtual environments. Naturally, this also holds true to immersive virtual environments. These environments can greatly improve perception of the virtual content over those provided by desktop displays, and allow users to use mid-air spatial input to directly interact with such content, mimicking physical interactions and making manipulations feel more natural. Nonetheless, interactions in mid-air also have drawbacks. In this thesis, we studied ways of overcoming some of these. In the next sections, we summarize the work conducted in this thesis, discuss its main results, and present directions for future research.

6.1. Dissertation Overview

Being of such relevance, virtual object manipulation has been subject of intensive research. To familiarize the reader with the context of our and previous research, we began by introducing relevant concepts for virtual environments in general, and for object manipulation in such environments in particular. Then, we surveyed

the most relevant manipulation techniques following several interaction paradigms, ranging from traditional mouse-based interfaces, to multi-touch approaches and mid-air techniques. A discussion of the presented state-of-the-art allowed us to identify trends and open challenges.

Furthermore, no generic solution could be derived from analyzing existing research, as manipulation methods have been proposed and tested for a variety of virtual environments, designed for different applications, and have different constraints given by visualization and tracking systems. We compared several techniques to manipulate objects in mid-air, based on the literature, in both semi and fully-immersive environments, through user evaluations. Additionally, we compared the performance and user preferences of those two kinds of virtual environments. Our findings suggest that, if no restrictions exist, the best approach for manipulating mid-air objects is to use a direct technique with an exact mapping, simultaneously controlling translation and rotation in 6-DOF, within a fully-immersive environment. These studies also revealed that mid-air manipulations lack precision, being difficult to place an object accurately in the desired position and orientation.

Focusing on increasing the precision of object manipulation in mid-air, we investigated if it can benefit from DOF separation, which has been proved useful in other interaction paradigms. For this purpose, we implemented a set of widgets to support single DOF control in mid-air, and evaluated it against two approaches. One follows an exact mappings, and the other dynamically scales users' movements. From this evaluation, we found that DOF separation can indeed improve precision, but at the cost of significantly higher completion times for more complex tasks. In order to combine the best aspects of each the evaluated techniques, we developed a novel technique, WISDOM, that combines widgets for DOF separation, scaled movements for translation, and direct manipulation for coarse transformations. However, after comparison with the original widgets technique and the direct approach, it presented no practical benefits, mostly due to the high number of available actions users were required to remember.

In a second attempt of reducing task times while offering a high level of precision, we conceived another mid-air manipulation technique, MAiOR. Depending on where users started interacting with an object, MAiOR allows them to translate or rotate the object in 3-DOF, which can be restrained to a single custom axis. If only coarse transformations are desired, MAiOR also offers direct manipulation that can be unlocked when in translation mode and, if highly accurate positioning is required, users can toggle a scaled translation mapping on and off. Although MAiOR did not showed significant improvements, mostly due to the metaphor used for rotations that

was classified as being difficult to use, we were able to draw valuable insights: separating translation and rotation transformations, more than single DOF control, is the most relevant aspect for offering precise manipulations and preventing unwanted errors; and, if an accurate tracking solution is available, scaling user movements does not provide significantly better placement precision.

Another challenge with immersive virtual environments, is the ability to interact with objects that are outside arms' reach. Direct approaches require users to grab the desired object while intersecting it with their hand. However, this selection method might not be feasible when objects are far from the user. The most common approaches require users to point at the desired object, but this can be ineffective due to the low accuracy related to mid-air gestures. We developed PRECIOUS, a new selection technique that implements an iterative progressive refinement strategy in immersive virtual environments. With a manipulable cone, users can select a group of objects, and are then moved closer to them for an easier selection. This refinement can be repeated until a single object is selected. Through a user evaluation, we compared PRECIOUS against two techniques from the literature, one follows a simple cone-casting approach, and the other employs an arm-stretching metaphor. Results revealed PRECIOUS as a versatile approach to out-of-reach target acquisition, combining accurate selection with consistent task completion times across different scenarios.

6.2. Conclusions and Discussion

The results we attained in the several user evaluations we carried out along this thesis allowed us to gather valuable considerations for designing effective mid-air manipulations of virtual objects. Regarding our initial assessments, we found that a technique such the Simple Virtual Hand [73], which is a direct approach that applies an exact mapping to control simultaneous 6-DOF, is the preferred method, ideally used within a fully-immersive environment. Nonetheless, if only positional tracking is available, or the interaction is being designed for a semi-immersive environment, the Handle-bar [116] technique can be better suited, as it uses only the position of both hands to control 7-DOF (translation, rotation and uniform scaling) in a way participants appreciated, and prevents objects being occluded by the users' hands.

After another user evaluation, we confirmed our first hypothesis, which stated that single DOF manipulation can contribute for a more precise placement than direct approaches. Indeed, our widgets implementation using virtual handles significantly reduced position and orientation error in docking tasks in mid-air, mostly due to successfully preventing unwanted transformations. This was possible because this technique isolated not only transformations, but also DOF within each transformation. Still, the baseline techniques used also revealed positive aspects: the Simple Virtual Hand was consistently fast, and the scaled translations of PRISM [43] were well perceived by participants. This led us to draw a tentative set of guidelines: (1) direct 6-DOF manipulation is suitable for fast and coarse transformations; (2) separating transformations helps prevent unexpected outcomes; (3) 1-DOF transformations are useful for fine adjustments; (4) scaled movements effectively reduce positioning error in translations.

By implementing those guidelines into two novel techniques, we were able to collect additional insights. Our objective was also to test our second and third hypothesis, which referred that DOF separation with scaled mappings can further increase placement precision, and that custom transformation axes could perform faster than using axes from object's or world's frames while maintain the same level of precision, respectively. We found, however, that no significant improvements came from scaling down isolated translations, and that participants did not sought 1-DOF control, having considered that simultaneously controlling the 3-DOF of a single transformation was enough for an accurate positioning. This disproves our hypothesis 2 and 3. As such, an improved set of guidelines for object manipulation within arms' reach, derived from our studies, are as follows:

- Simultaneous 6-DOF control with exact mappings is the best choice considering only naturalness and celerity, as it is suited just for coarse transformations;
- If an accurate object placement is desired, the most important feature that should be supported is DOF separation through isolated transformations;
- Single-DOF control and scaled user movements appear to be useful only to reduce errors cause by jitter from imprecise user tracking systems.

The techniques we developed showed that it is possible to achieve millimetric accuracy, as long as a reliable tracking is used, an objective that was slightly higher then the very difficult task used by Frees and Kessler [42]. To go further than that, into sub-millimetric precision, more than manipulation techniques and tracking solutions need to be considered, as those distances are difficult to be perceived. In such cases, combining our findings with the viewpoint adjustment proposed by Osawa [99],

which makes objects and distances appear larger, might lead to the intended accuracy. However, an approach like this needs to be carefully implemented, in order to prevent undesired side effects, such as cybersickness or loss of context.

Regarding interactions with objects placed outside arms' reach, we showed that iterative progressive refinement strategies can be used in IVEs to execute accurate selections, thus verifying our fourth hypothesis. Actually, moving users closer to the objects, through an instantaneous teleport action, is an effective alternative to the zoom approaches used in non-immersive scenarios [9], as it eases selection without causing any discomfort to users.

Our work has its limitations nonetheless. The proposed techniques did not show as much benefits as we hoped. This, in part, was caused by the difficulty felt by the participants in the user evaluations in recalling all the available transformations. As some of them stated, they would need more time to properly take advantage of the techniques. Unfortunately, due to time constrains, we could not evaluate the learning effect. We believe that, with experience, our techniques would perform better. This would make them more suited for professional settings, but such requirement makes them less interesting for common users. These techniques also had support for both uniform and 1-DOF scaling, which is uncommon for mid-air techniques, as we saw in the surveyed literature. In order to better focus our work, and due to the lack of proper baselines, this transformation lacked a proper performance evaluation. Lastly, while our technique for out-of-reach selections was evaluated in an environment with some distractors, we did not test it in heavily cluttered environments. We believe such scenarios will bear different results, making our technique to perform in a not so effective manner. For that cases, it might be needed to disrupt user immersion with objects' rearrangement techniques, bearing in mind that they might not deal well with multiple similar objects.

6.3. Future Work

Despite the work we presented in this dissertation, interacting with objects in mid-air within immersive virtual objects still offers avenues for future research. In the next paragraphs we summarize some of these possible directions.

As we reported, MAiOR's rotation approach was generally classified as hard to grasp for participants not acquainted with 3D software. We consider the need for metaphors that are easy to learn and use for isolated 3-DOF rotations as one of

the main open challenges for precise manipulations in mid-air. People are used to control translation and rotation together in everyday life, which allows them to easily fix an incorrect positioning revealed after a rotation transformation. Having these two transformations separated hinders this process. We believe that one way that this can be mitigated is using pivot points for rotation, as already proposed for sketch-based interfaces [109]. Also, our findings regarding 3-DOF versus 1-DOF contradicts suggestions from previous research [125]. As such, further testing should be carried out in order to clarify this.

When discussing existing mid-air manipulation techniques, we noticed that scaling is often disregarded. However, this transformation is often grouped together with translation and rotation in software for 3D content creation and editing. Because of this, we proposed approaches for applying scaling transformations both uniformly and according to each axis independently. As mentioned before, the scaling performance of our techniques was not properly evaluated in this work. A careful evaluation of these techniques can give additional insights to eventually lead to the creation of novel techniques that encompass the three transformations.

Regarding out-of-reach interactions, we believe that an increased starting cone's reach in our proposed technique can significantly reduce selection times. It would be interesting to assess if that modification is enough to achieve times similar to Flashlight, while keeping the very low number of incorrect selections. Extending it to be better suited for heavily cluttered environments, using different or additional refinement mechanisms, is also worth of attention. For instance, a better approach to the final refinement step could take into account the actual position of objects, and use some heuristics to determine whether a disambiguation grid should be used, instead of always showing it when two objects are selected. Additionally, combining our technique with a manipulation approach for out-of-reach, such as HOMER [20, 133], can create an all around technique capable of interacting with objects at any distance.

Finally, we reckon that interactions with virtual objects in mid-air can benefit from novel interaction paradigms, resorting to approaches based not only on software, but also on specific hardware. Examples of topics that might be worth exploring are voice and haptics. These can be used, for instance, to add additional modalities for input and feedback to aid in faster and more precise object manipulations in immersive virtual environments.

6.4. Final Remarks

In conclusion, and with all things considered, we have validated our thesis. We have shown that hyper-natural approaches, such as DOF separation and iterative progressive refinement strategies, can successfully be used to provide more effective mid-air interactions within immersive virtual environments.

We hope that our insights can help researchers, interaction designers and developers to create better manipulation techniques. Reducing the impact of the lack of precision in mid-air interactions can lead to interfaces that allow us to take more of the increasingly common immersive virtual environments. Of course, others aspects need to be addressed, such as the fatigue that mid-air gestures and longer standing periods can cause, in order to make these immersive settings more useful and productive and, eventually, to take the place of the decades-old desktop interfaces.

Bibliography

[1] L. Aguerreche, T. Duval, and A. Lécuyer. 3-hand manipulation of virtual objects. In *Proceedings of the 15th Joint virtual reality Eurographics conference on Virtual Environments*, pages 153–156. Eurographics Association, 2009.

[2] B. R. D. Araújo, G. Casiez, and J. A. Jorge. Mockup builder: direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, pages 173–180. Canadian Information Processing Society, 2012.

[3] B. R. D. Araujo, G. Casiez, J. A. Jorge, and M. Hachet. Mockup builder: 3d modeling on and above the surface. *Computers & Graphics*, 37(3):165 – 178, 2013.

[4] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013.

[5] O. K.-C. Au, C.-L. Tai, and H. Fu. Multitouch gestures for constrained transformation of 3d objects. In *Computer Graphics Forum*, volume 31, pages 651–660. Wiley Online Library, 2012.

[6] C. Auteri, M. Guerra, and S. Frees. Increasing precision for extended reach 3d manipulation. *The International Journal of Virtual Reality*, 12(1):66–73, 2013.

[7] A. S. Azevedo, J. Jorge, and P. Campos. Combining eeg data with place and plausibility responses as an approach to measuring presence in outdoor virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 23(4):354–368, 2014.

[8] M. Azimi. Skeletal joint smoothing white paper. Technical report, Microsoft, 2012. Online: `http://msdn.microsoft.com/en-us/library/jj131429.aspx`, accessed 5-January-2018.

[9] F. Bacim, R. Kopper, and D. A. Bowman. Design and evaluation of 3d selection techniques based on progressive refinement. *International Journal of Human-Computer Studies*, 71(7):785–802, 2013.

[10] H. Benko and S. Feiner. Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. In *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, page 22, 2007.

[11] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. R. Blum, and J. R. Cooperstock. Did minority report get it wrong? superiority of the mouse over 3d input devices in a 3d placement task. In *IFIP Conference on Human-Computer Interaction*, pages 400–414. Springer, 2009.

[12] L.-P. Bergé, E. Dubois, and M. Raynal. Design and evaluation of an "around the smartphone" technique for 3d manipulations on distant display. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 69–78, New York, NY, USA, 2015. ACM.

[13] F. Bettio, A. Giachetti, E. Gobbetti, F. Marton, and G. Pintore. A practical vision based approach to unencumbered direct spatial manipulation in virtual worlds. In *Eurographics Italian Chapter Conference*, pages 145–150, 2007.

[14] A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–370. ACM, 2005.

[15] E. A. Bier. Snap-dragging in three dimensions. *ACM SIGGRAPH Computer Graphics*, 24(2):193–204, 1990.

[16] B. Bollensdorff, U. Hahne, and M. Alexa. The effect of perspective projection in multi-touch 3d interaction. In *Proceedings of Graphics Interface 2012*, GI '12, pages 165–172, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[17] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 14(3):262–270, July 1980.

[18] B. Bossavit, A. Marzo, O. Ardaiz, L. D. De Cerio, and A. Pina. Design choices and their implications for 3d mid-air manipulation techniques. *Presence: Teleoper. Virtual Environ.*, 23(4):377–392, Nov. 2014.

[19] D. Bowman and R. P. McMahan. Virtual reality: how much immersion is enough? *Computer*, 40(7):36–43, 2007.

[20] D. A. Bowman and L. 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*, pages 35–ff. ACM, 1997.

[21] D. A. Bowman and L. F. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages & Computing*, 10(1):37–53, 1999.

[22] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. An introduction to 3-d user interface design. *Presence: Teleoperators and virtual environments*, 10(1):96–108, 2001.

[23] D. A. Bowman, R. P. McMahan, and E. D. Ragan. Questioning naturalism in 3d user interfaces. *Communications of the ACM*, 55(9):78–88, 2012.

[24] G. Bruder, F. Steinicke, and W. Stuerzlinger. Effects of visual conflicts on 3d selection task performance in stereoscopic display environments. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE Press, 2013.

[25] F. M. Caputo and A. Giachetti. Evaluation of basic object manipulation modes for low-cost immersive virtual reality. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*, pages 74–77. ACM, 2015.

[26] J. Cashion, C. Wingrave, and J. J. LaViola Jr. Dense and dynamic 3d selection for game-based virtual environments. *IEEE transactions on visualization and computer graphics*, 18(4):634–642, 2012.

[27] L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung. Touching the void: direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2625–2634, New York, NY, USA, 2010. ACM.

[28] E. Chapoulie, M. Marchal, E. Dimara, M. Roussou, J.-C. Lombardo, and G. Drettakis. Evaluation of direct manipulation using finger tracking for complex tasks in an immersive cube. *Virtual Reality*, 18(3):203–217, 2014.

[29] I. Cho and Z. Wartell. Evaluation of a bimanual simultaneous 7dof interaction technique in virtual environments. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 133–136. IEEE, 2015.

[30] A. Cohé, F. Dècle, and M. Hachet. tbox: A 3d transformation widget designed for touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3005–3008, New York, NY, USA, 2011. ACM.

[31] A. Cohé and M. Hachet. Understanding user gestures for manipulating 3d objects from touchscreen inputs. In *Proceedings of Graphics Interface 2012*, pages 157–164, Toronto, Canada, May 2012. ACM.

[32] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. Van Dam. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 183–188. ACM, 1992.

[33] L. D. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114. ACM, 1997.

[34] F. Daiber, E. Falk, and A. Krüger. Balloon selection revisited: multi-touch selection techniques for stereoscopic data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 441–444, New York, NY, USA, 2012. ACM.

[35] B. A. Davis, K. Bryla, and P. A. Benton. *Oculus Rift in action*. Manning, 2015.

[36] S. Davis, K. Nesbitt, and E. Nalivaiko. Comparing the onset of cybersickness using the oculus rift and two virtual roller coasters. In *Proceedings of the 11th Australasian Conference on Interactive Entertainment (IE 2015)*, volume 27, page 30, 2015.

[37] H. G. Debarba, J. G. Grandi, A. Maciel, L. Nedel, and R. Boulic. Disambiguation canvas: a precise selection technique for virtual environments. In *IFIP Conference on Human-Computer Interaction*, pages 388–405. Springer, 2013.

[38] L. Dipietro, A. M. Sabatini, and P. Dario. A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):461–482, 2008.

[39] J. Feng, I. Cho, and Z. Wartell. Comparison of device-based, one and two-handed 7dof manipulation techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pages 2–9. ACM, 2015.

[40] A. Fernandes and S. Feiner. Combating vr sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2016.

[41] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 95–96. ACM, 1996.

[42] S. Frees and G. D. Kessler. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 99–106. IEEE, 2005.

[43] S. Frees, G. D. Kessler, and E. Kay. Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1):2, 2007.

[44] B. Froehlich. The quest for intuitive 3d input devices. *HCI International, Las Vegas, Nevada USA*, 2005.

[45] B. Froehlich, J. Hochstrate, V. Skuk, and A. Huckauf. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 191–199. ACM, 2006.

[46] B. Fröhlich and J. Plate. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 526–531. ACM, 2000.

[47] A. Giesler, D. Valkov, and K. Hinrichs. Void shadows: Multi-touch interaction with stereoscopic objects on the tabletop. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*, SUI '14, pages 104–112, New York, NY, USA, 2014. ACM.

[48] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2006.

[49] T. Grossman and D. Wigdor. Going deeper: a taxonomy of 3d on the tabletop. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on*, pages 137–144. IEEE, 2007.

[50] J. Guerreiro, D. Medeiros, D. Mendes, M. Sousa, J. Jorge, A. Raposo, and I. Santos. Beyond post-it: structured multimedia annotations for collaborative ves. In *Proc. of the Eurographics Conference on Virtual Environments*, pages 55–62, 2014.

[51] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

[52] M. Hachet, B. Bossavit, A. Cohé, and J.-B. de la Rivière. Toucheo: Multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 587–592, New York, NY, USA, 2011. ACM.

[53] M. Hachet, P. Guitton, and P. Reuter. The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 225–112. ACM, 2003.

[54] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM, 2005.

[55] M. Hancock, S. Carpendale, and A. 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, pages 1147–1156, New York, NY, USA, 2007. ACM.

[56] M. Hancock, T. ten Cate, and S. 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, pages 133–140, New York, NY, USA, 2009. ACM.

[57] M. Hancock, F. Vernier, D. Wigdor, S. Carpendale, and C. Shen. Rotation and translation mechanisms for tabletop interaction. In *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on*, pages 8 pp.–, 2006.

[58] O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. 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, 2009. ACM.

[59] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson. Holodesk: Direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2421–2430, New York, NY, USA, 2012. ACM.

[60] S. Houde. Iterative design of an interface for easy 3-d direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 135–142, New York, NY, USA, 1992. ACM.

[61] C. E. Hughes, L. Zhang, J. P. Schulze, E. Edelstein, and E. Macagno. Cavecad: Architectural design in the cave. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 193–194. IEEE, 2013.

[62] P. Issartel, F. Guéniat, T. Isenberg, and M. Ammi. Analysis of locally coupled 3d manipulation mappings based on mobile device motion. *CoRR*, abs/1603.07462, 2016.

[63] N. Katzakis and M. Hori. Mobile devices as multi-dof controllers. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 139–140. IEEE, 2010.

[64] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93. ACM, 2001.

[65] T. Kim and J. Park. 3d object manipulation using virtual handles with a grabbing metaphor. *IEEE Computer Graphics and Applications*, 34(3):30–38, May 2014.

[66] K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of*

Graphics interface 2009, pages 119–124. Canadian Information Processing Society, 2009.

[67] K. Kin, T. Miller, B. Bollensdorff, T. DeRose, B. Hartmann, and M. Agrawala. Eden: A professional multitouch tool for constructing virtual organic environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352. ACM, 2011.

[68] K. Kiyokawa, H. Takemura, and N. Yokoya. Manipulation aid for two-handed 3-d designing within a shared virtual environment. In *HCI (2)*, pages 937–940, 1997.

[69] S. Knoedel and M. Hachet. Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces*, 3DUI '11, pages 75–78, Washington, DC, USA, 2011. IEEE Computer Society.

[70] R. Kopper, F. Bacim, D. Bowman, et al. Rapid and accurate 3d selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 67–74. IEEE, 2011.

[71] R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 601–610. ACM, 2005.

[72] J. LaViola Jr. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1), 2000.

[73] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. A. Bowman, and I. Poupyrev. *3D User Interfaces: Theory and Practice (2nd Edition)*. Addison-Wesley, 2017.

[74] J. Liang and M. Green. Jdcad: A highly interactive 3d modeling system. *Computers & graphics*, 18(4):499–506, 1994.

[75] J. Liu, O. K.-C. Au, H. Fu, and C.-L. Tai. Two-finger gestures for 6dof manipulation of 3d objects. *Computer Graphics Forum*, 31(7):2047–2055, 2012.

[76] P. Lubos, G. Bruder, and F. Steinicke. Analysis of direct selection in head-mounted display environments. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 11–18. IEEE, 2014.

[77] J. F. Lucas, D. A. Bowman, J. Chen, and C. A. Wingrave. Design and evaluation of 3d multiple object selection techniques. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005.

[78] L. S. M. *Virtual Reality*. Cambridge University Press, 2017.

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

[80] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III*, INTERACT'11, pages 461–476, Berlin, Heidelberg, 2011. Springer-Verlag.

[81] A. Martinet, G. Casiez, and L. Grisoni. The design and evaluation of 3d positioning techniques for multi-touch displays. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces*, pages 115–118, 2010.

[82] A. Martinet, G. Casiez, and L. 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, 2010. ACM.

[83] M. McGuire and M. Mara. The G3D innovation engine, 01 2017. `https://casual-effects.com/g3d/`.

[84] D. Medeiros, F. Carvalho, L. Teixeira, P. Braz, A. Raposo, and I. Santos. Proposal and evaluation of a tablet-based tool for 3d virtual environments. *SBC*, 4(2):31, 2013.

[85] D. Medeiros, E. Cordeiro, D. Mendes, M. Sousa, A. Raposo, A. Ferreira, and J. Jorge. Effects of speed and transitions on target-based travel techniques. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 327–328, New York, NY, USA, 2016. ACM.

[86] D. Medeiros, L. Teixeira, F. Carvalho, I. Santos, and A. Raposo. A tablet-based 3d interaction tool for virtual engineering environments. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 211–218. ACM, 2013.

[87] D. Mendes and A. Ferreira. Evaluation of 3d object manipulation on multi-touch surfaces using unconstrained viewing angles. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part IV*, pages 523–526. Springer, 2011.

[88] D. Mendes, P. Lopes, and A. Ferreira. Hands-on interactive tabletop lego application. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE '11, pages 19:1–19:8, New York, NY, USA, 2011. ACM.

[89] M. Mine, A. Yoganandan, and D. Coffey. Making vr work: building a real-world immersive modeling application in the virtual world. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pages 80–89. ACM, 2014.

[90] M. Mine, A. Yoganandan, and D. Coffey. Principles, interactions and devices for real-world immersive modeling. *Computers & Graphics*, 48:84–98, 2015.

[91] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM Press/Addison-Wesley Publishing Co., 1997.

[92] M. Moehring and B. Froehlich. Effective manipulation of virtual objects within arm's reach. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 131–138. IEEE, 2011.

[93] M. Möllers, P. Zimmer, and J. Borchers. Direct manipulation and the third dimension: Co-planar dragging on 3d displays. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 11–20, New York, NY, USA, 2012. ACM.

[94] T. S. Mujber, T. Szecsi, and M. S. Hashmi. Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology*, 155:1834–1838, 2004.

[95] M. A. Nacenta, P. Baudisch, H. Benko, and A. Wilson. Separability of spatial manipulations in multi-touch interfaces. In *Proceedings of Graphics Interface 2009*, GI '09, pages 175–182, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

[96] T. T. H. Nguyen and T. Duval. Poster: 3-point++: A new technique for 3d manipulation of virtual objects. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 165–166. IEEE, 2013.

[97] T. T. H. Nguyen, T. Duval, and C. Pontonnier. A new direct manipulation technique for immersive 3d virtual environments. In *ICAT-EGVE 2014: the 24th International Conference on Artificial Reality and Telexistence and the 19th Eurographics Symposium on Virtual Environments*, page 8, 2014.

[98] G. M. Nielson and D. R. Olsen Jr. Direct manipulation techniques for 3d objects using 2d locator devices. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 175–182. ACM, 1987.

[99] N. Osawa. Two-handed and one-handed techniques for precise and efficient manipulation in immersive virtual environments. In *Advances in Visual Computing*, pages 987–997. Springer, 2008.

[100] G. Perelman, M. Serrano, M. Raynal, C. Picard, M. Derras, and E. Dubois. The roly-poly mouse: Designing a rolling input device unifying 2d and 3d interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 327–336. ACM, 2015.

[101] F. Periverzov and H. Ilies. Ids: The intent driven selection method for natural user interfaces. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 121–128, March 2015.

[102] J. S. Pierce, B. C. Stearns, and R. Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 141–145. ACM, 1999.

[103] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. 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*, pages 79–80. ACM, 1996.

[104] I. Poupyrev and T. Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages & Computing*, 10(1):19–35, 1999.

[105] M. Prachyabrued and C. W. Borst. Visual feedback for virtual grasping. In *3D User Interfaces (3DUI) 2014*. IEEE, 2014.

[106] E. Ragan, A. Wood, R. McMahan, and D. Bowman. Trade-offs related to travel techniques and level of display fidelity in virtual data-analysis environments. In *ICAT/EGVE/EuroVR*, 2012.

[107] J. L. Reisman, P. L. Davidson, and J. 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, 2009. ACM.

[108] W. Robinett and R. Holloway. Implementation of flying, scaling and grabbing in virtual worlds. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 189–192. ACM, 1992.

[109] R. Schmidt, K. Singh, and R. Balakrishnan. Sketching and composing widgets for 3d manipulation. *Computer Graphics Forum*, 27(2):301–310, 2008.

[110] U. Schultheis, J. Jerald, F. Toledo, A. Yoganandan, and P. Mlyniec. Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3d tasks. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 117–124. IEEE, 2012.

[111] K. Shoemake. Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*, volume 92, pages 151–156, 1992.

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

[113] A. L. Simeone, A. Bulling, J. Alexander, and H. Gellersen. Three-point interaction: combining bi-manual direct touch with gaze. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 168–175. ACM, 2016.

[114] S. Smith and T. Marsh. Evaluating design guidelines for reducing user disorientation in a desktop virtual environment. *Virtual Reality*, 8(1), 2004.

[115] R. So, W. Lo, and A. Ho. Effects of navigation speed on motion sickness caused by an immersive virtual environment. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 43(3), 2001.

[116] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1297–1306, New York, NY, USA, 2012. ACM.

[117] M. Sousa, D. Mendes, R. K. D. Anjos, D. Medeiros, A. Ferreira, A. Raposo, J. M. Pereira, and J. Jorge. Creepy tracker toolkit for context-aware interfaces. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pages 191–200. ACM, 2017.

[118] M. Speicher, F. Daiber, S. Gehring, and A. Krüger. Exploring 3d manipulation on large stereoscopic displays. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays*, pages 59–66. ACM, 2016.

[119] A. Steed and C. Parker. 3d selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th International Immersive Projection Technology Workshop*, pages 13–14, 2004.

[120] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995.

[121] S. Strothoff, D. Valkov, and K. 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, 2011. ACM.

[122] E. Suma, S. Finkelstein, M. Reid, S. Babu, A. Ulinski, and L. Hodges. Evaluation of the cognitive effects of travel technique in complex real and virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 16(4), 2010.

[123] C. Telkenaroglu and T. Capin. Dual-finger 3d interaction techniques for mobile devices. *Personal and ubiquitous computing*, 17(7):1551–1572, 2013.

[124] A. Van Dam, D. H. Laidlaw, and R. M. Simpson. Experiments in immersive virtual reality for scientific visualization. *Computers & Graphics*, 26(4):535–555, 2002.

[125] M. Veit, A. Capobianco, and D. 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*, pages 51–58. ACM, 2009.

[126] D. Vogel and P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 657–666. ACM, 2007.

[127] V. Vuibert, W. Stuerzlinger, and J. R. Cooperstock. Evaluation of docking task performance using mid-air interaction techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pages 44–52. ACM, 2015.

[128] R. Wang, S. Paris, and J. Popović. 6d hands: Markerless hand-tracking for computer aided design. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 549–558, New York, NY, USA, 2011. ACM.

[129] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 63:1–63:8, New York, NY, USA, 2009. ACM.

[130] Y. Wang, C. L. MacKenzie, V. A. Summers, and K. 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*, pages 312–319. ACM Press/Addison-Wesley Publishing Co., 1998.

[131] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 269–278. ACM, 2007.

[132] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2011.

[133] C. Wilkes and D. A. Bowman. Advantages of velocity-based scaling for distant 3d manipulation. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 23–29. ACM, 2008.

[134] A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 67–76. ACM, 2008.

[135] S. Wu, A. Chellali, S. Otmane, and G. 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*, pages 59–68. ACM, 2015.

[136] R. C. Zeleznik, A. S. Forsberg, and P. S. Strauss. Two pointer input for 3d interaction. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 115–120. ACM, 1997.

[137] S. Zhai and P. Milgram. Human performance evaluation of manipulation schemes in virtual environments. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 155–161. IEEE, 1993.

A

Spatial User Tracking with Multiple Depth Cameras

To develop the prototypes used in the user evaluations of Sections 4.1, 4.2 and 5.1, we used our own user tracking solution, the *Creepy Tracker*. It consists of a network server that combines data from multiple depth sensors to provide full-body positional tracking of people within a room-sized volume. It automatically selects the best sensor to follow each person, handling occlusions and maximizing interaction space, while providing full-body tracking in scalable and extensible manners. It also keeps position and orientation of stationary interactive surfaces, while offering continuously updated point-cloud user representations combining both depth and color data. A performance evaluation showed that, although slightly less precise than marker-based optical systems, *Creepy Tracker* provides reliable multi-joint tracking without any wearable markers.

In this appendix, we introduce *Creepy Tracker*'s main concepts and explain how it functions. Although it has support for several features, here we will only present those that

are relevant for the work of this thesis. This is a subset of our paper [117], where the full description of the *Creepy Tracker* can be found.

A.1. Overview

The *Creepy Tracker* toolkit uses a network of distributed sensor units connected to a central hub. Each sensor unit is composed of a Microsoft Kinect depth camera and a standalone C# application running on a single computer. The number of sensors is directly related to the area required by the interaction being designed. Interactions with a single typical (up to 4×2m) vertical surface may require one or two units, while interactions around a tabletop most commonly need several (up to 5) sensors surrounding that surface. Each sensor unit provides a continuous data stream. These converge on the tracker's central hub, which is responsible for synchronization, processing and merging the data, as depicted in Figure A.1. The central hub also broadcasts the state of the tracked environment to client applications. Moreover, for the virtual model of the tracked people and surfaces to be precisely aligned with the physical topology of the room, the sensors' position and orientation must be first calibrated. Adding surfaces requires an active calibration for each new surface by defining the surface plane using 3D depth data of one sensor. After calibration, as people move in the tracked area, the virtual model gets updated in real-time, while broadcasting the updated data.

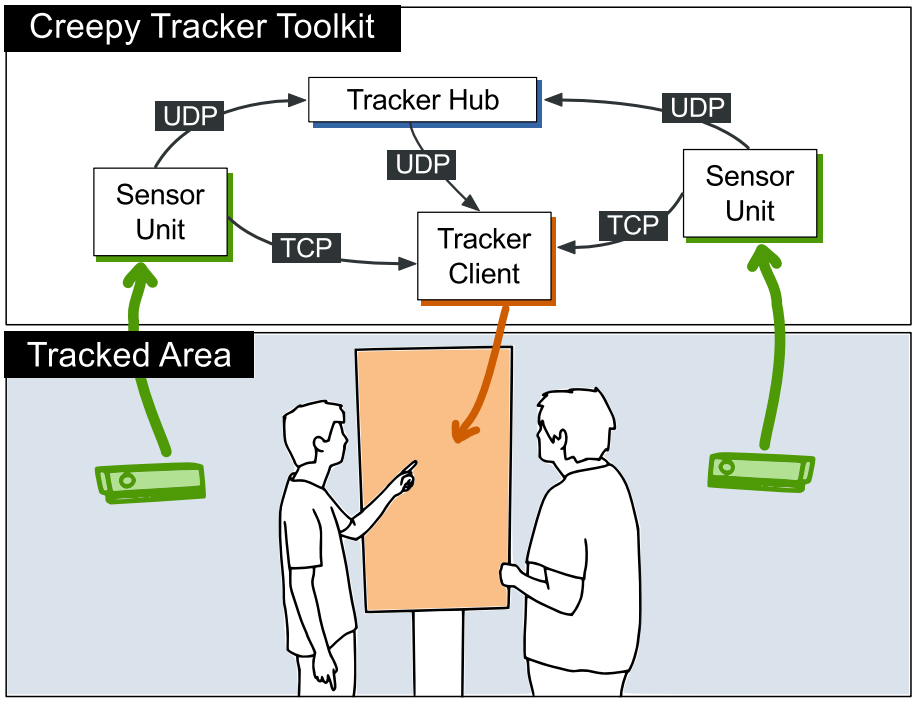


Figure A.1: Overall system's architecture.

A.2. Sensor Unit

All sensor units, each of them connected to an individual depth sensor, capture color, depth data and the body tracking model of every observed person in the tracked area. Each body model is associated with a numerical factor to represent the estimated degree of confidence about the quality of the tracked person, which is sent together with the body model data. The confidence factor is calculated by adding all tracked body joints' weight, while discarding inferred ones. The weight of each joint can be customizable, so that the tracker can favor specific joints, useful for different scenarios. For instance, pointing tasks require far more importance given to hands' than feet' joints. Figure A.2 shows an individual sensor client tracking two people, the person closer to the camera has a lower degree of confidence because half of the lower limbs' joints cannot be seen. Tracked people with confidence factors below a configurable threshold are ignored. Color and depth data are processed for the point-cloud representation of each person. The body tracking model is broadcast to the Tracker Hub using an UDP stream, while point-clouds are available via a concurrent TCP connection.

Figure A.2: *Creepy Tracker*'s sensor unit.

A.3. Tracker Hub

The Tracker Hub component handles the unified model of the tracked area by combining the data streams from all sensor units. To create a reliable model, the Tracker Hub requires a calibration process to transform all received data into a single coordinate system. Figure A.3 shows calibrated sensors with both position and orientation matching the

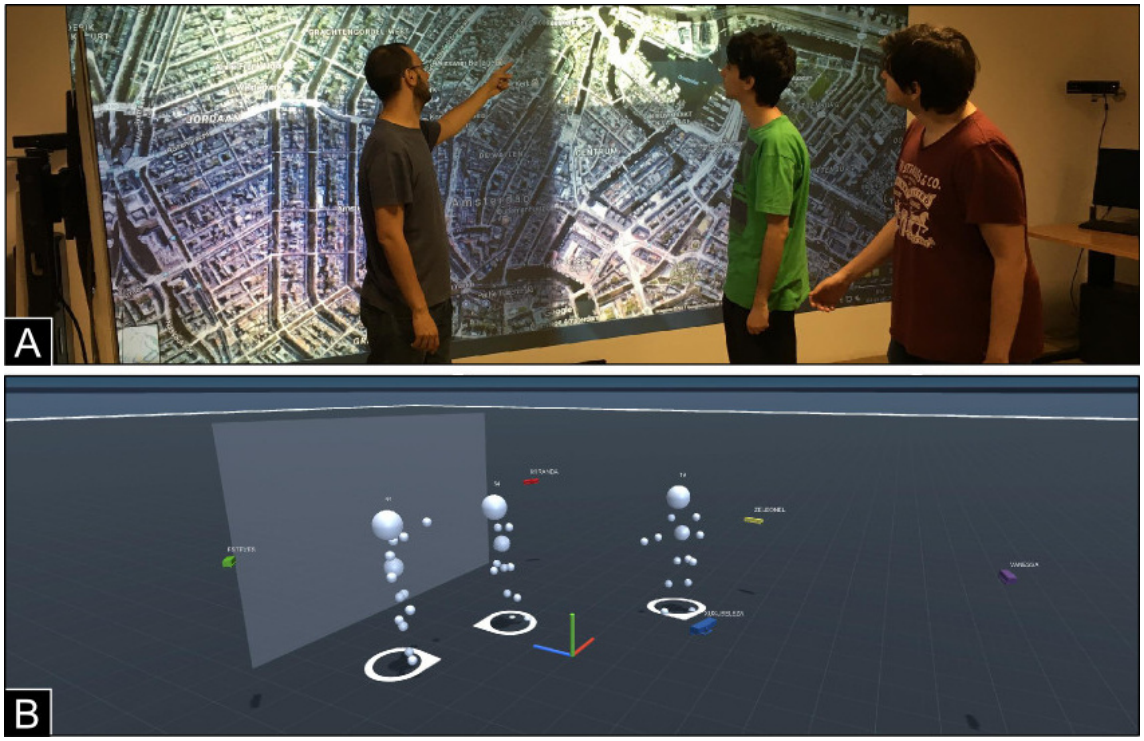


Figure A.3: A physical setting with depth sensors, users and a surface (A), and its corresponding representation in the *Creepy Tracker*'s hub.

physical cameras. Data received from each of those sensor units will be spatially correct in the unified model's coordinate system. Analogously, a surface calibrated on one depth camera's coordinate system also is transformed to match the area of the physical one, as shown in the same figure. Since a surface is a collection of four fixed points in space, the setting up and calibration process needs to occur only once. The Tracker Hub is a Unity3D application that acts as broadcast server of the unified model to application clients.

A.4. Calibration Method

A calibration process is required to unify all data streams into a single coordinate system. For this *Creepy Tracker* relies on the body tracking model of a person from each sensor unit to calculate the new global coordinate system and all cameras' position and orientation. The calibration process requires one standing person to be seen by all sensor units, in two discrete steps. Figure A.4A shows five uncalibrated sensor units before the calibration process.

Creepy Tracker requires calibration parameters from body tracking models at two distinct locations separated by the distance of a step to calculate the origin and forward and up vectors of the new calibrated coordinate system. In the first step, the position of the person

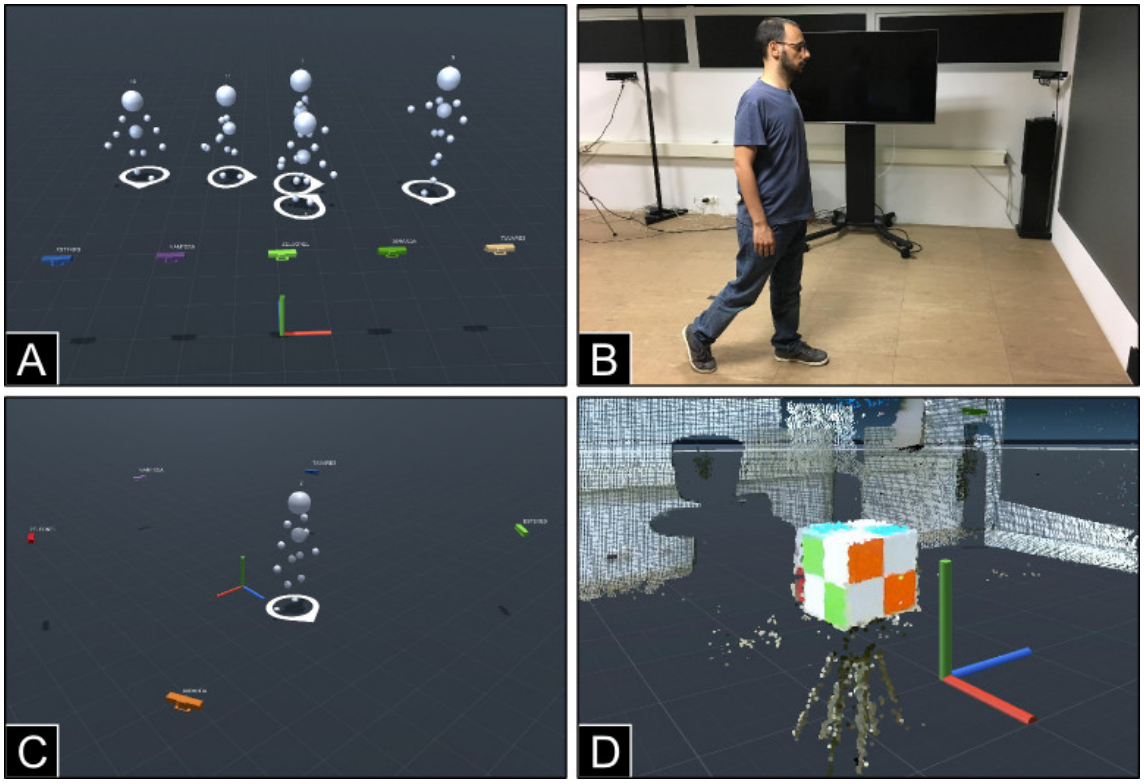


Figure A.4: Calibration process: (A) center; (B) step forward; (C) result; and (D) calibration cube for manual adjustments.

is used to define the origin. The up vector, defined by the spine base and spine shoulder joints of the body model, is also stored, as well as the position of both feet. The second calibration step can be performed after the person moves a step forward (Figure A.4B). This new position is used in conjunction with the first to define the coordinate system forward vector. The second up vector is averaged with the first to minimize the impact of incorrect poses when calculating the final up vector. Finally, the minimum height according to the up vector of the four feet positions is used to define floor's position. Figure A.4C shows five calibrated sensors around the coordinate system's origin.

This calibration is usually enough for most interactive scenarios. However, we reckon that a more precise calibration might be needed for more demanding cases. For such situations, we created an additional calibration step. It consists of capturing a depth data frame of each sensors and displaying them using point-clouds, with a simple object placed in the middle of the tracked area. For this, we resort to a cardboard cube with a coloured checkerboard in each face (Figure A.4D). Then, is possible to manually adjust position and rotation parameters of each sensor, so that the point clouds match as well as possible. A new calibration process is required when the setup undergo any adjustment or modification in sensor units.

A.5. Tracking People

The Tracker Hub formalizes a body tracking model from the sensor unit into a *Body* entity and a person into an instance of *Human*. Overlapping individual body tracking models from different sensor units map into a single person. Consequently, a *Human* preserves a set of *Bodies*, one for each seen by a sensor unit.

When information regarding a new *Body* arrives, the Tracker Hub will try to fit that body in a *Human* within a parameterizable distance threshold. This threshold is set by default to 30 cm, to account for different sensors' perspectives, as it is impossible for two people to have their Spine Base joints closer than this threshold without intimate space violations. The distance is calculated according to Spine Base joints of both *Bodies*. If there is no suited *Human*, a new one is created. When a *Body* is no longer seen by its sensor, it is dissociated from the corresponding *Human*. If a *Human* has no more associated *Bodies*, it will enter a waiting period of 1 second. During that period, if a new *Body* appears within the distance threshold from the *Human*'s last position, it is associated to that *Human*, which exits from the waiting period. Otherwise, if no *Body* is associated with the *Human* until the waiting period expires, the *Human* is removed from the tracker.

Each *Human* entity is constantly choosing the most appropriate *Body* by selecting the one with the highest confidence value. It is not always easy to acknowledge for sure where people are turned to, as some sensors may be facing each other and perceiving mirrored body models for the same person, because the Microsoft Kinect cannot distinguish between people facing forwards or backwards. To overcome this, we follow two approaches. Firstly, we consider a disambiguation pose consisting of having at least one forearm approximately parallel to the floor. The direction one is pointing at, can be used to define that person's forward vector, as it would be both unnatural and very difficult to accomplish such a pose with the arm pointing backwards. When this vector's direction is opposite from the current *Human*'s forward, we automatically mirror *Body*'s left and right data. Secondly, as front and back switching occurs mainly when a *Body* from a different sensor is chosen, we also mirror the *Body* when the *Human* is detected to rotate faster (approximately 180 degrees in two consecutive frames) than it is humanely possible.

To deal with the known noisy skeleton information from the Microsoft Kinect, we implemented a double exponential smoothing filter [8]. The filter's parameters can be configured to achieve a compromise between smoothness and added latency. This filter is applied to *Human*'s joints, not to *Bodies*, and helps when dealing with sensor switching in setups with coarse calibrations.

A.6. Using Tracker Data

Creepy Tracker offers a client-side C# API with a layer to render network communication transparent and provide updated encapsulated abstractions of tracking data. An independent tracker client, upon connection, continuously receives a list of *Humans* and can request at any time a list of available *Surfaces*. A *Human* is a representation of a real person in tracked area. It holds an unique identification provided by the tracker, a point in space correspondent to the person's position, a client-side calculation of the person's direction and a list of all body joints.

B

Choosing a Target-based Travel Technique

Travel plays an essential part on experience in VE, where the user moves from a starting point A to a target point B. We can divide travel in two subcategories: *Explore*, where users move freely on the VE without a predetermined goal, and *Search*, where they have to reach a specific checkpoint. The choice of the travel technique can influence the user and cause severe side effects, such as cybersickness [72], reduced presence and disorientation [114].

The more natural the technique, the more efficiently users can perform travelling tasks on VEs [122], especially on Explore tasks. However, constraints such as fatigue and limitations of the physical space can make it unsuitable to some situations. Indirect methods such as Target-based and Steering techniques [73] can overcome this problem by providing an approach to travel while still providing a favorable spatial orientation on VEs.

Some causes of cybersickness in IVEs include graphical realism of the environment [36], field-of-view [40] and navigation speed [115]. Although steering techniques can provide an improved spatial understanding of virtual surroundings, target-based approaches can reliably overcome unwanted symptoms on inexperienced users of immersive systems [106].

In this work, we investigate the effects of speed and transition in Target-based techniques, as presented in our paper [85]. We compared three different methods and how they impact the experience in key aspects such as comfort and cybersickness.

B.1. Techniques Implemented

We implemented three different techniques for travel in IVEs, as depicted in Figure B.1.

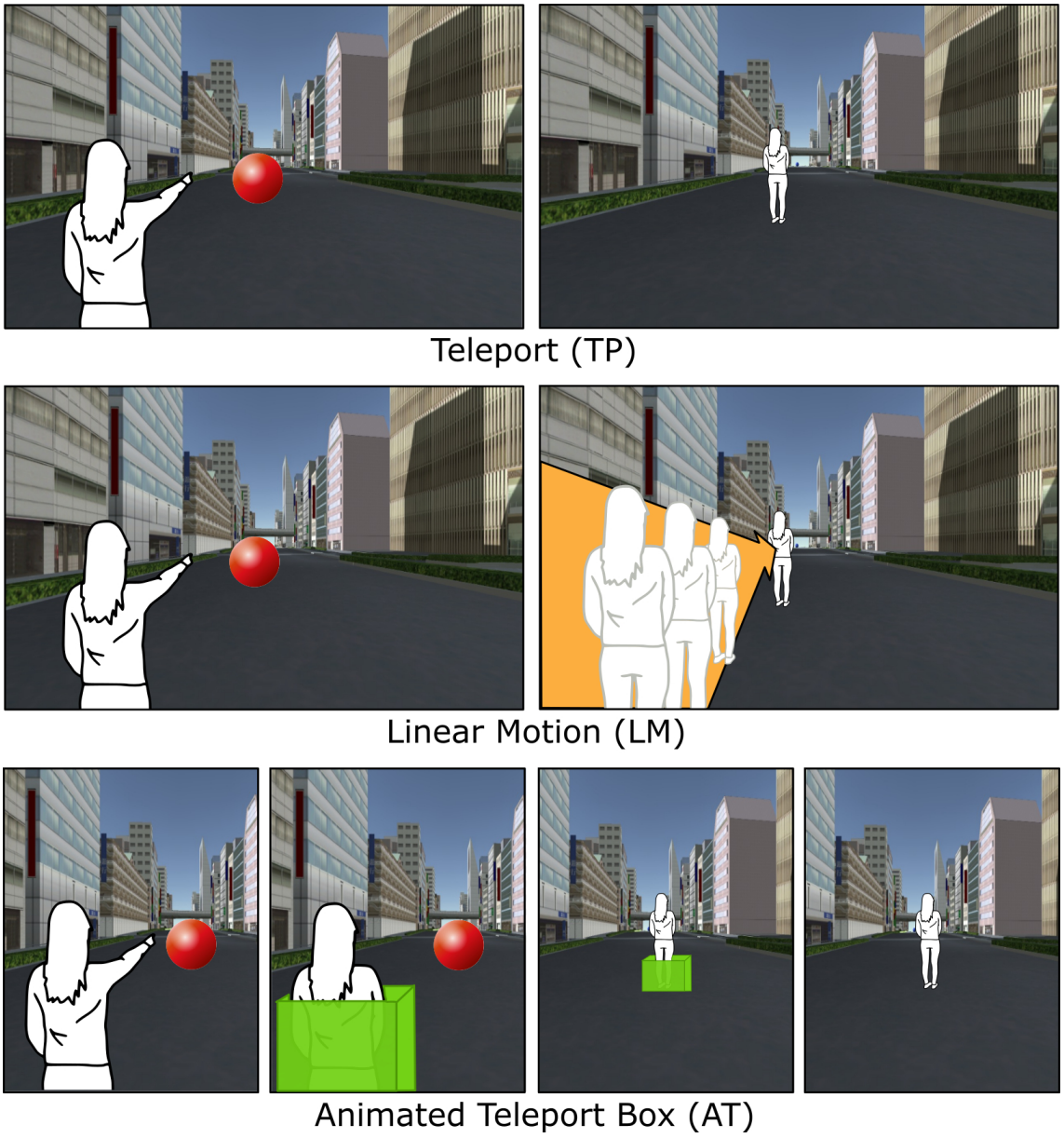


Figure B.1: Implemented Travel techniques.

B.1.1. Teleport

The Teleport technique [73] (TP), also known as infinite velocity, translates a person instantaneously from their current position to the next checkpoint.

B.1.2. Linear Motion

The Linear Motion technique (LM) consists of moving the user along a linear path for two seconds with a constant velocity, until the next checkpoint.The velocity choice is based on previous work [115] and varies between 30 m/s and 50 m/s depending on the checkpoint distance.

B.1.3. Animated Teleport Box

We developed the Animated Teleport Box (AT) technique with the objective to combat the negative effects of the Teleport technique. Two 1.5 second animations were played when a user was being translated from their current position to next checkpoint. The first one animated the Box to rise up and surround the user, and the second one executed the same animation but in the inverse direction. The box has 2.3 meters on each side so that users would not feel too claustrophobic when travelling. It was developed with the intention of not showing users that they were being moved, as a mean of decreasing the disorientation that might be felt after being teleported.

B.2. User Evaluation

To validate the techniques described above, we completed a user evaluation. Our aim was to understand which of the techniques were preferred and the impact of cybersickness on users. We tested the techniques in our laboratory in a controlled environment, using a Samsung GearVR HMD with a Samsung Galaxy S7 smartphone. Users were able to freely rotate their head within the VE. 20 participants (two females) completed the user evaluation, with ages ranging from 19 to 31 years old (average: 24) and seven participants already had previous experience in VR. Each user evaluation session adopted the same protocol, starting the initial briefing with a quick explanation to the experiment and also with a description of the techniques. To avoid biased results from users becoming famil-

iarized with the techniques and used to the environment, the techniques were presented in a partial random order, so all permutations were exhausted.

The virtual environment was a model of the city of Osaka, Japan (visible in Figure B.1), which was populated with six spherical checkpoints to where the users would be travelling to. During each travel, the users were told where the next checkpoint would be (to their left or right) and were also instructed to point to said checkpoint before traveling using the techniques. Users had no control over the path they would take, and would only be in charge of pointing to the checkpoints. We allowed the users an adjustment period to the environment, before travelling to the first checkpoint, to make sure they knew where they were and where they were being moved to. Each session took on average thirty minutes, which ended with a brief questionnaire about their experience.

B.3. Results and Discussion

Throughout data analysis, we first conducted a Shapiro-Wilk test which showed that not all samples followed a normal distribution. We then used a Friedman non-parametric test to look for statistical significance between the three tested techniques. When statistical differences were found, we conducted a Wilcoxon Signed-Ranks Test to look for statistical significance on each pair of techniques using the Bonferroni correction.

For a better comparison regarding task performance, we subtracted the animation times from the total time following the formula: $T' = T - \alpha \times (n - 1)$, where T is the total time, α the path time (3 seconds in AT, 2 in LM, and zero in TP) and n the number of travels (6 in our case). Looking at the chart from Figure B.2, we can notice a slightly

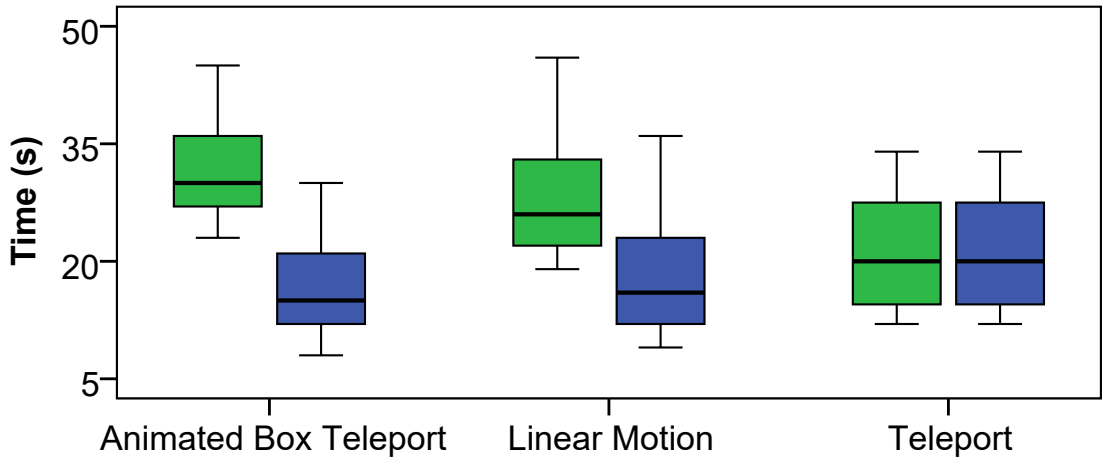


Figure B.2: Time elapsed on each task. Green box-plots represent total time, and blue the time excluding techniques' animations.

Question	**AT**	**LM**	**TP**
It was easy	5 (1)	5 (1)	5 (1)
It was satisfying	4 (2)	4.5 (2)	4 (2)
I felt physical discomfort *	1 (1)	2 (3)	1 (1)
I felt visual discomfort	1 (1)	2 (2)	1 (1)

Table B.1: User preferences: Median (Interquartile Range). Higher median values express accordance to the statement, and * indicates statistical significance.

better performance with AT, but without statistical significance. Because of that we can state that efficiency is similar in all the tested techniques.

Regarding questionnaires' data (Table B.1), we found that users felt more physical discomfort using LM ($Z = -2.699$, $p < 0.0005$ against AT, and $Z = -2.386$, $p = 0.017$ against TP). Despite the discomfort caused by LM, participants stated it as their favourite technique in most cases.

Due to the similarity between user preferences on both AT and TP we conducted an additional test on the total times of the test task. This test confirms a better result on such condition with TP as it does not need additional time among the movement between positions ($Z = -3.114$, $p < 0.0005$ against AT and $Z = -2.578$, $p = 0.01$ against LM).

In short, we found that Infinite Velocity techniques cause less discomfort. We also found that using transition effects in conjunction with these techniques affect neither performance nor cybersickness.