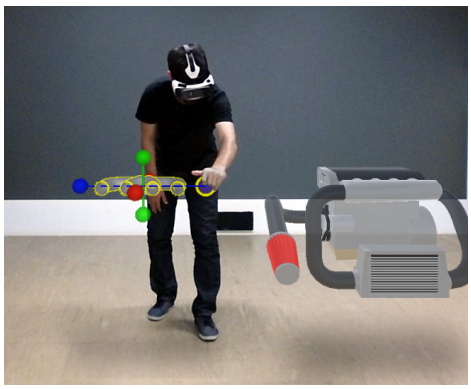
TÉCNICO LISBOA



DOF Separation in Mid-air 3D Object Manipulation

Filipe Miguel da Silva Reis de Miranda Relvas

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Alfredo Manuel dos Santos Ferreira Júnior
Eng. Daniel Filipe Martins Tavares Mendes

Examination Committee

Chairperson: Prof. João António Madeiras Pereira
Supervisor: Prof. Alfredo Manuel dos Santos Ferreira Júnior
Member of the Committee: Prof. Carlos Alberto Pacheco dos Anjos Duarte

November 2016

Acknowledgments

First of all I would like to thank Instituto Superior Técnico and the VIMMI group of INESC-ID Lisboa, for giving me the opportunity to carry out this work. To Prof. Dr. Alfredo Ferreira for this demanding but rewarding challenge, and all of his help and advising throughout the year.

To Daniel Mendes, for his incredible help and overall support, not only as a co-advisor but as a friend. I can't thank you enough.

A special thanks to my thesis brother in arms, Eduardo Cordeiro. Without your help, criticism and healthy discussions I couldn't have done it.

To all the people that took their precious time to participate in several user evaluations, especially Diogo Melo for his availability in such a short notice.

I would also like to mention the financial support from the Portuguese Foundation for Science and Technology (FCT), through projects TECTON3D (ref. PTDC/EEI-SII/3154/2012) and IT-MEDEX (ref. PTDC/EEISII/6038/2014).

To my friends, Rafaela Faria for being always there through my ups and downs, Miguel Andrade for his questionable jokes and gaming sessions and Lourenço Silva for our intense discussions about unnecessary but insightful topics.

And last but not least, to my parents, grandparents and my brother for supporting me so much during this hard but incredible journey. Without them this would not have been possible.

Resumo

Manipulação de objetos é uma funcionalidade chave em quase todos os ambientes virtuais. No entanto é difícil manipular objetos de forma precisa em ambientes virtuais imersivos usando gestos no ar que replicam interações no mundo físico, apesar de ser uma abordagem natural e direta. Estudos anteriores concluíram que a separação de graus de liberdade em interfaces táteis levaram a resultados melhorados.

Neste trabalho, avaliamos o impacto da separação de 6 DOF explícita em tarefas de manipulação no ar. Implementámos uma técnica baseada em Widgets virtuais que permite o controlo de um DOF de cada vez, e comparámo-la com uma abordagem direta e a técnica PRISM, que ajusta de forma dinâmica o rácio entre o movimento da mão e do objeto. Os resultados sugerem que a separação de DOF beneficia a precisão em manipulações no espaço à custa de tempos mais longos em tarefas complexas. Destes resultados elaborámos alguns pontos para futura orientação em manipulação de objetos no ar.

Com base nestes pontos desenvolvemos uma técnica de nome WISDOM, que combina os aspetos positivos das técnicas estudadas anteriormente, dispondo de manipulação direta e movimentos escalados mas mantendo a separação de DOF oferecida pela nossa implementação de Widgets. De forma a validar a nossa proposta, conduzimos outra avaliação com utilizadores de forma a compará-la com 6DOF e Widgets. Apesar do WISDOM não reduzir os tempos de conclusão das tarefas, melhorou o erro na posição e rotação em tarefas que requerem apenas translação ou rotação e em tarefas mais complexas que requerem ambas.

Palavras-chave: Interfaces de Utilizador 3D, Ambientes Virtuais Imersivos, Interações no Espaço, Manipulação de Objetos no Ar, Separação de Graus de Liberdade

Abstract

Object manipulation is a key feature in almost every virtual environment. However, it is difficult to accurately place an object in immersive virtual environments using mid-air gestures that mimic interactions in the physical world, although being a direct and natural approach. Previous research studied mouse and touch based interfaces concluding that separation of degrees-of-freedom (DOF) led to improved results.

In this work, we assess the impact of explicit 6 DOF separation in mid-air manipulation tasks. We implemented a technique based on familiar virtual widgets that allow single DOF control, and compared it against a direct approach and PRISM, which dynamically adjusts the ratio between hand and object motions. Our results suggest that full DOF separation benefits precision in spatial manipulations, at the cost of additional time for complex tasks. From these results we drew some guidelines for object manipulation in mid-air.

Based on the guidelines we developed a technique, WISDOM, that combined the positive aspects of the three techniques used before. Our technique provides direct manipulation and scaled movements while maintaining DOF separation offered by our Widgets implementation. In order to validate our proposal we conducted another user evaluation to compare it against 6DOF and Widgets. While WISDOM didn't reduce task time completion, it improved position and rotation error in translation and rotation only tasks and more complex tasks that required both translation and rotation simultaneously compared to the other techniques.

Keywords: 3D User Interfaces, Immersive Virtual Environments, Spatial Interactions, Mid-air Object Manipulation, DOF Separation

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We are currently witnessing a huge interest in virtual reality (VR), mainly due to the recent technological advances that made head-mounted displays (HMD) affordable and widely available. The overall investment in these technologies has increased in the last few years not only from a hardware perspective but also from an application one. Immersive virtual environments (IVE) that were made possible with such technologies are being used for several purposes: Healthcare where VR as an affordable tool could provide significant benefits for individuals with disabilities [1] or just a means for understanding, assessing and treating a number of clinical mental disorders [2]; Gaming and Entertainment with Playstation VR, Oculus Rift and HTC Vive, affordable and easy to use head-mounted displays to improve game immersion; Architecture and Engineering where such technology can act as a possible complement/substitute to the existent CAD software [3]. Being of such importance, object manipulation in virtual environments has been subject of research for long, covering different kinds of interaction paradigms. Mouse and touch interfaces for instance, benefited from degrees-of-freedom separation (DOF). Previous user evaluation showed better performance when compared to direct approaches, mainly due to the required mapping between the 2D input and 3D output.

1.1 Challenge

To interact within immersive virtual environments (IVE), the ability to manipulate virtual objects is a key feature. In mid-air interactions the dimensional difference between input and output does not exist. While direct approaches that mimic interactions in the physical world are the most natural, it is still difficult to place a virtual object in the desired place with a high degree 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. With this work we intend to answer the following research questions:

1. Do single DOF transformations reduce position and rotation error in mid-air manipulations?

2. Can single DOF manipulations in mid-air be as fast as direct approaches?

1.2 Approach

In order to validate and confirm our question, related to precise object manipulation, we started by studying the state of art regarding traditional mouse-based manipulations, multi-touch surfaces and stereoscopic tabletops interaction, and mid-air object manipulation. Based on existing literature we conducted an initial evaluation to gather information on the impact of DOF separation in mid-air object manipulation, followed by a comparative study between three techniques. The first technique followed a direct approach, the second scaled users' movement and the third was our implementation of mid-air virtual handles for DOF separation. We then carried out a user evaluation where each user was asked to perform a set of docking tasks with different degrees of complexity. From the attained results, we drew guidelines for future mid-air object manipulation techniques and proposed a new technique, WISDOM (Widgets combining Scaled movements and DOF separation for Object Manipulation). The technique gathered the positive aspects of each one of the three techniques initially studied namely scaled movement, DOF separation and direct object manipulation. We also conducted tests for our proposal in order to validate the proposed technique.

1.3 Contributions

The development of the work described in this document resulted in following contributions:

- **Assessment of DOF separation in mid-air**

  We started this work by comparing three manipulation techniques in order to evaluate the importance of DOF separation in mid-air. This assessment resulted in a user evaluation and can be seen as important for future work regarding object manipulation in VEs.

- **Object manipulation technique guidelines**

  As a result of our assessment evaluation, we were able to draw some guidelines for object manipulation in IVEs. These should aid researchers and developers in creating better techniques that can combine the better aspects of each evaluated approach.

- **An object manipulation technique**

  Our proposed approach, combines the strongest aspects of each one of the three techniques implemented for our initial assessment. The result is a technique that provides scaled movement, combined with DOF separation and direct object manipulation. It is also the first technique to offer 1DOF, 2DOF and 3DOF object scaling in mid-air.

1.4 Publications

The work developed for this dissertation resulted in the following publications:

1. Daniel Mendes, **Filipe Relvas**, Alfredo Ferreira e Joaquim A. Jorge, *The Benefits of DOF Separation in Mid-air 3D Object Manipulation*, VRST 2016 - The 22nd ACM Symposium on Virtual Reality Software and Technology, November 2016.

2. **Filipe Relvas**, Daniel Mendes, Alfredo Ferreira e Joaquim A. Jorge, *Separação de Graus de Liberdade em Manipulação no Espaço de Objectos 3D*, EPCGI'2016 - 23º Encontro Português de Computação Gráfica e Interação, November 2016.

1.5 Thesis Outline

In the remainder of the document, we survey the most relevant related work in Chapter 2, then we define our prototype architecture and setup in Chapter 3. The assessment of DOF separation in mid-air is detailed in Chapter 4, followed by our proposal's description in Chapter 5. Evaluation and validation is depicted in Chapter 6. Finally, we present our conclusions and point out directions for future work in Chapter 7.

Chapter 2

Related Work

Techniques for manipulating virtual objects have been subject of research in the past few decades. From mouse to mid-air, passing through touch enabled surfaces, several approaches have been proposed, ever trying to more effectively position and orient objects in virtual environments. As such we categorize previous work in traditional mouse-based manipulations, interacting with multi-touch surfaces, manipulation on stereoscopic tabletops and mid-air manipulation.

2.1 Traditional Mouse-Based Manipulations

To overcome the difference between input and output DOF, most mouse-based techniques for virtual object manipulation rely on some sort of widgets, which reduce the simultaneous DOF being controlled. Houde [4] proposed the handle box (Fig. 2.1), a bounding box surrounding the object being manipulated, with a lifting handle to move the object up and down, and four rotation handles, to rotate the object about its central axis. Drag is performed by clicking the object inside the handle box. Conner et al. [5] also resorted to virtual handles to develop 3D widgets for performing 9DOF transformations on virtual objects (Fig. 2.2). The handles are used to constrain geometric transformations to a single plane or axis. Dragging one of the handles translates, rotates or scales the object depending on which mouse button is pressed. For rotations, the direction of the user's initial gesture determines which of the two

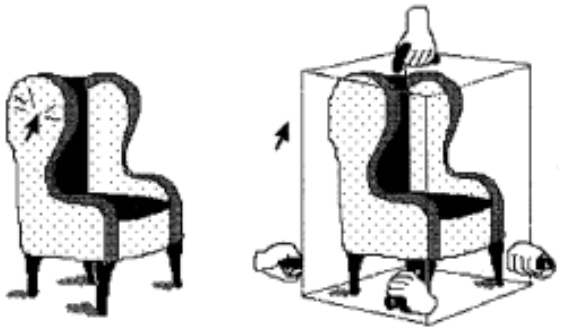
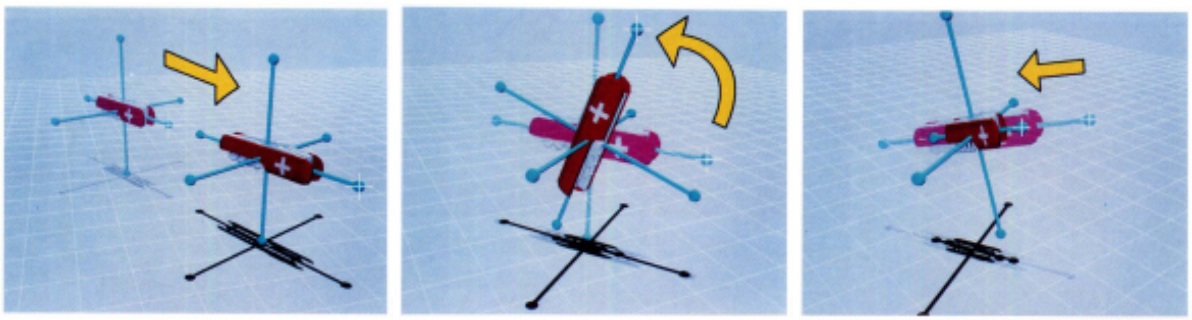


Figure 2.1: Handle Box [4]: (Left) Sliding, (Right) Lifting and rotating.



(a) Translation. (b) Rotation. (c) Scale.

Figure 2.2: Three-dimensional Widgets technique [5].

axes perpendicular to the handle is used as rotation's axis. Focusing only in rotations, Ken Shoemake proposed Arcball [6], where users can draw an arc on the screen projection of a sphere to change object's orientation. These techniques attained such popularity, they are still used in common commercial applications for creating or editing 3D virtual models, like Unity3D[1] or SketchUp[2].

2.2 Using Interactive Surfaces

Distinctly from mouse techniques, interactive surfaces allow users to directly touch objects displayed. Schmidt et al. [7] presented a novel approach to 3D manipulation, using a mixture of 3D widgets, context sensitive suggestions and gestural commands (Fig. 2.3). The user indicates an object to transform by explicitly selecting it with a tap of a pen. After selecting the desired object the user draws a stroke, and the system responds by automatically creating translation and rotation widgets based on the candidate axis nearest to the stroke. These initial widgets can be modified using context-sensitive gestures, or by simply drawing another axis. Cross gesture for instance corresponds to an apply or select action, whereas a double cross gesture indicates that the crossed target should become persistent. The au-

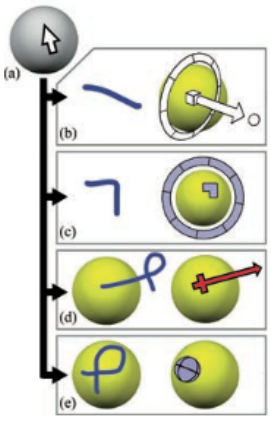


Figure 2.3: Sketched and composed Widgets [7]: (a) Selection, (b) Rotate/translate/scale widget, (c) Screen-space widget, (d) Transformation stroke and (e) Pivot.

[1] Unity3D: `http://unity3d.com`, last visited June 30th 2016.
[2] SketchUp: `http://sketchup.com`, last visited June 30th 2016.

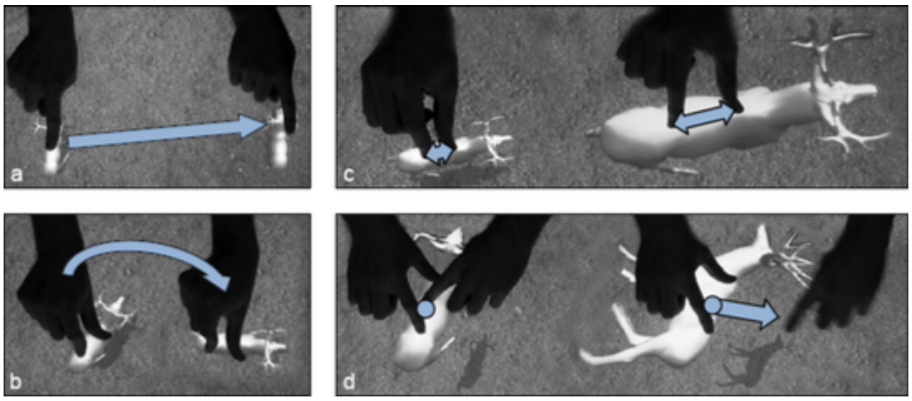


Figure 2.4: Sticky Fingers technique [8]: (a) Move, (b) Rotate 2D, (c) Lift, (d) Rotate 3D.

thors noted that unconstrained 3D manipulation is difficult for inexperienced users and the axis-dragging widget provided unsatisfactory results since it's difficult to draw a rotation axis parallel to the view vector. Discoverability problems may also arise causing user confusion.

Force-based interaction was the approach taken by Mark Hancock et al. [8], with Sticky Tools, a combination of three concepts. The first concept, Sticky Fingers (Fig. 2.4), consists on interacting with a virtual object, by keeping the same point of contact through the whole interaction. Translations along the XX and YY axes can be performed with one finger while rotating two fingers relative to one another applies a rotation to the virtual object around the Z axis. It is also possible to lift the object by increasing or decreasing the distance between both fingers. In order to achieve rotations around the XX and YY axes the second concept, Opposable Thumb, was introduced by authors. It makes use of a third finger as relative input, providing rotation about the axis orthogonal to the direction of movement, enabling full 6DOF control. Lastly Virtual Tools' idea is to assign richer meaning to virtual objects, thus providing more complete functionality to tabletop interfaces. Virtual objects can now be understood as nouns, verbs, reconfigurable tools, attributes or pure objects allowing people to pick objects up, place them in other objects or simply use them as tools. A physics engine is also in place, so these virtual objects interact with each other while keeping contact with the user finger. By using physical forces user interaction gets easier thanks to our ability to transfer knowledge from physical world experiences to virtual ones.

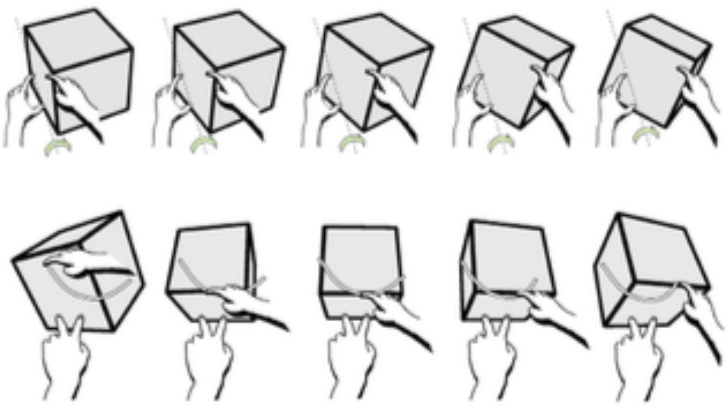


Figure 2.5: Screen-space formulation for 2D and 3D [9].

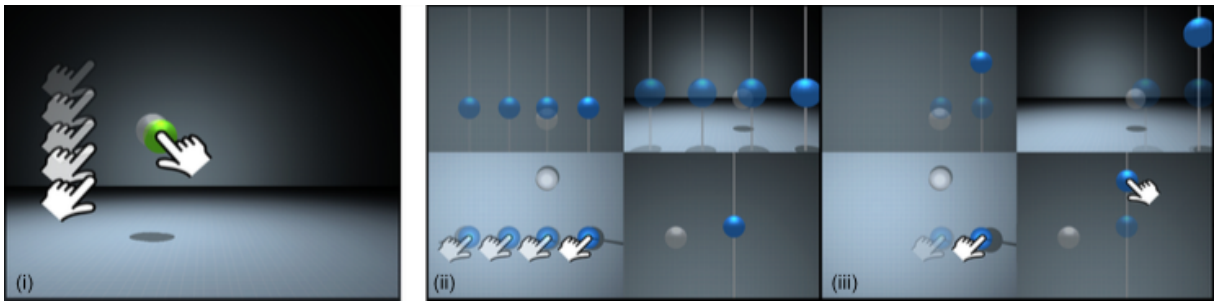


Figure 2.6: DOF separation in 3D manipulation tasks [10]: (i) Illustration of the Z-Technique, (ii, iii) Illustration of the multi-touch viewport technique.

Jason L. Reisman et al. [9] presented a screen-space method which allows direct control in 2D and 3D on a multi-touch surface (Fig. 2.5). This method defines each contact point as a constraint, which ensures the screen-space projection of the object-space point touched always remains underneath the user's fingertip. Mapping is updated regularly, so that constraints are best met, and TRS (Translate-Rotate-Scale) semantics for 2D manipulation were kept unchanged. However due to screen-space limitations problems arise, such as ambiguous rotations, rotational exhaustion or just general movement barriers. The authors believe these disadvantages are generic and apply to any screen-space 3D direct manipulator.

Reduction of simultaneous DOF controlled have been later suggested by Anthony Martinet et al. [10] with the presentation of two techniques (Fig. 2.6). The first one, multi-touch viewport, is an extension to the standard four view-ports technique found in commercial CAD (Computer-Aided Design) applications. The screen is divided into four equally sized viewports and each viewport uses the same field of view for its virtual camera. When a finger first touches the surface, the corresponding viewport is detected and a ray orthogonal to the view is projected into the scene, where eventually an object will be intersected. Subsequent finger movements move the object in the plane parallel to the view passing through the object center. When a second finger touches another viewport, the multi-touch viewport extension is activated. The second technique presented by the authors, called the Z-Technique, enables the user to place an object at any arbitrary depth position. The object selection is achieved with ray-casting, coming from the camera center through the finger position, while a second indirect finger (not in contact with the object) provides depth position control by measuring backward-forward movement relative to the user position. These two techniques did not have a significant difference for completion time, multi-touch viewport performed well for low accuracy tasks while Z-Technique is a good alternative for small display surfaces.

Following their previous work Anthony Martinet et al. [11] proposed a taxonomy for 3D manipulation techniques with multi-touch displays. Using that taxonomy they introduced a new technique, DS3 (Depth-Separated Screen Space), where translation and rotation input is separated. By combining the Z-Technique [10] with the use of the constraint solver by [9], both described above, objects can be translated along the screen plane with one finger, while the second indirect finger provides depth position. To control object orientation two fingers directly in contact with the object are needed. This with the help of the constrain solver enables a clear separation between translation and rotation.

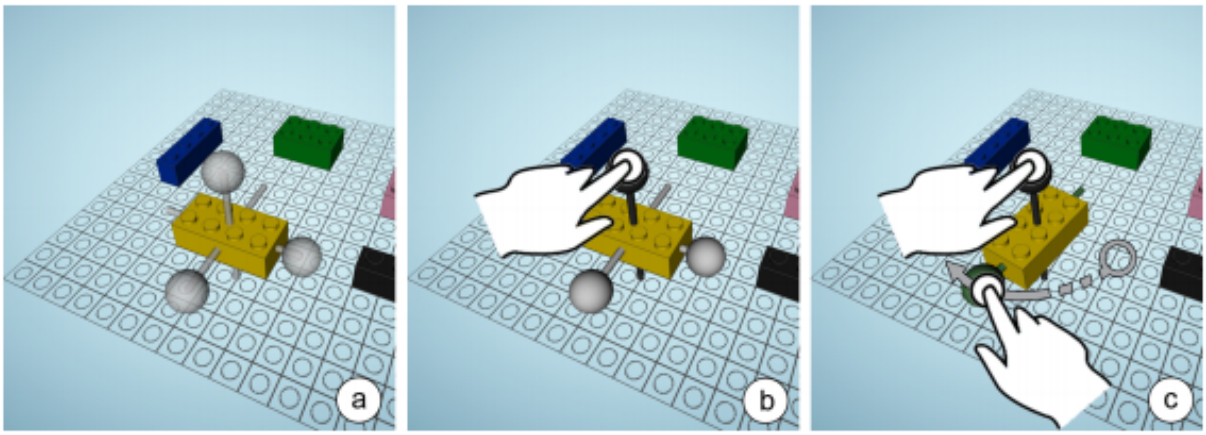


Figure 2.7: Rotation-Handles technique for object rotation [12]: (a) Active handles, (b) Selected rotation axis, (c) Performed rotation.

To create virtual environments for computer-animated films, Kin et al. [14] designed and developed Eden, a fully functional multi touch set construction application. Virtual objects can be translated using a direct drag approach. XX and YY axes translation is performed by touching the object with two fingers next to each other (conjoined touch), while Z axis translation is triggered with a conjoined touch together with a single finger touch drag up and down, thus separating horizontal and vertical movements. A single finger touch followed by a drag movement enables a virtual widget similar to the Arcball to apply object rotations. LTouchIt [12], although using direct manipulation for translations, also relies on widgets for rotations. Following the DOF separation, authors developed 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 (Fig. 2.7). This technique consists on three cylinders representing the possible rotation axes. Users can select one rotation axis by touching the sphere on the cylinder end. After selecting the desired axis a second touch followed by a second sphere drag performs an object rotation. The center of the object is considered the rotation center.

The tBox [13], a 3D transformation widget that appears as a wireframe box around the object to manipulate (Fig. 2.8), favors independent 9DOF control (three for translation, three for rotation and three for scale). Grabbing one of the box's edges followed by a drag movement in the direction of the edge triggers an object translation. Object rotation consists on dragging one of the box's faces in the desired direction. For object scaling, users need to drag two different edges from the same axes or from different

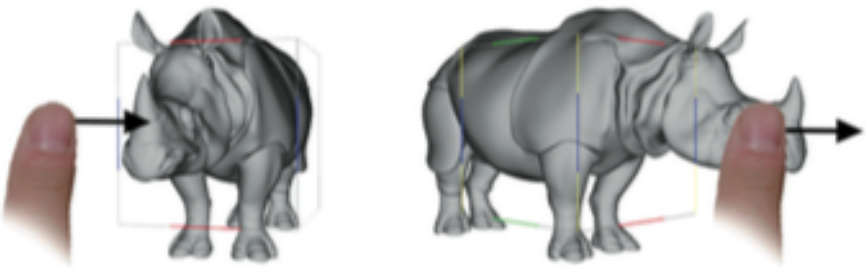


Figure 2.8: tBox 3D transformation Widget [13].

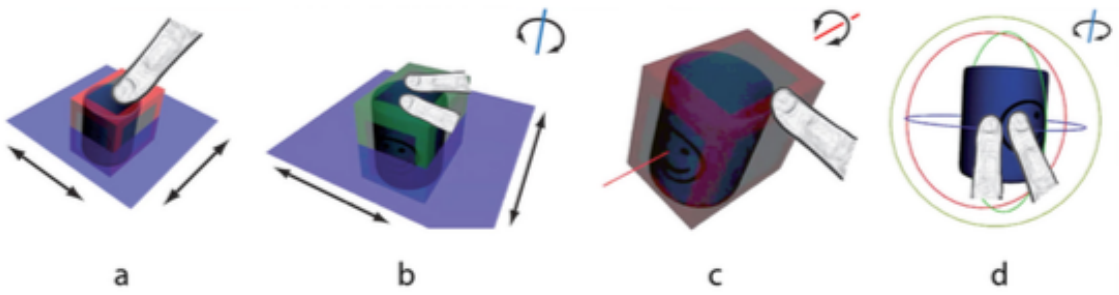


Figure 2.9: Gimbal Box interaction methods [15]: (a) In-Plane Translation, (b) In-Plane Translation and Rotation, (c) Axis Rotation, (d) Extended Arcball.

axes, to either scale uniformly or bi-directionally. Bollensdroff et al. [15] also developed a cube shaped widget, the Gimbal Box (Fig. 2.9), which uses a touch in one of its faces to translate in the plane defined by that face. To rotate the object the widget has two variations. One uses the TRS applied to a cube's face or, alternatively, touching an edge of the box induces a rotation around an axis parallel to the edge. The other variation is based on the Arcball. Through a controlled study, authors concluded that adapted widgets are superior to other approaches for multi-touch interactions.

It's clear that DOF separation provides an efficient yet predictable way to manipulate objects in a multi-touch environment. However the use of different transformation widgets that directly associate different operations, requires careful touch positioning to trigger the appropriate tools. Oscar Kin-Chung Au et al. [16] work makes use of the extra input bandwidth of multi-touch screens and delegates the manipulation power of standard transformation widgets to the multi-touch gestures. This enables seamless control of constraint and transformation manipulation using a single multi-touch action. The authors presented four techniques, starting by axis selection and transformation manipulation, where a candidate axis is selected with two touch points and the object is translated by keeping and moving two fingers. The rotation is also achieved by the gesture of two-finger pan, but perpendicular to the selected axis. The amount of translation and rotation is calculated according to the average distance between the current

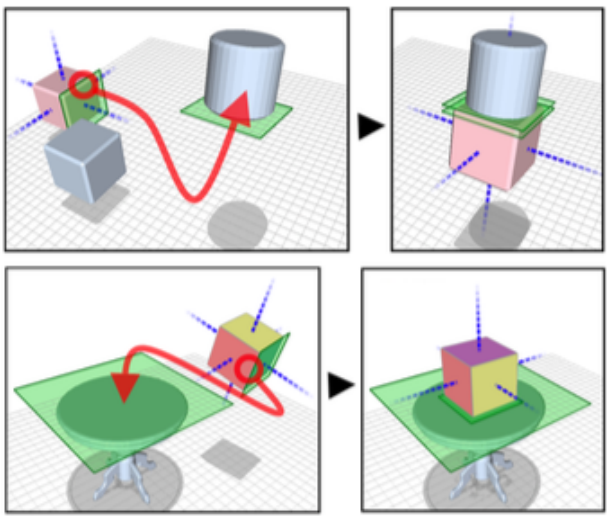
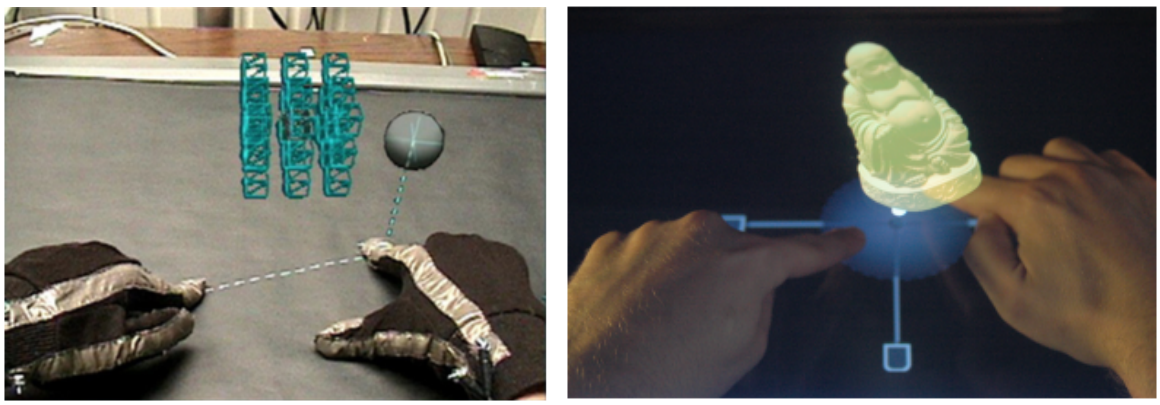


Figure 2.10: Multitouch gestures [16]: Active snapping.

and the initial contact points. Scaling is also featured and is designed as the gesture of two-finger pinch along the selected axis, no matter if the two fingers are pinched together or apart. The amount of scaling is defined by the ratio between the current and initial distances of the two contact points. The remaining two techniques are relative manipulations, active snapping (Fig. 2.10) automatically glues two objects such that their position and/or orientation are aligned, and axis and center borrowing where users can use another object axes to manipulate the selected one.

2.3 Manipulations on Stereoscopic Tabletops

Using semi-immersive displays, stereoscopic tabletops allow users to see both the physical and virtual world. In this case, since imagery appears on a volumetric space, interacting with virtual content presents different challenges. Benko and Feiner [17] decomposed 3DOF tasks into a set of 2DOF and 1DOF tasks, using a balloon metaphor (Fig. 2.11(a)). This metaphor was inspired by how people play with a helium balloon on a string. A tethered helium balloon floats straight up from the point where its string is being held. If one holds the string tightly with one hand, with the string passing loosely through the fingers of the remaining hand and moves the hands relative to each other, the balloon will change in height. Besides the need of specifically designed gloves, the authors concluded that such approach only works well for static objects and 3DOF tasks. Similarly, Strothoff et al. [19] proposed an approach to select and manipulate a cursor in stereoscopic tabletops, using two fingers to define a triangle, with a spherical cursor in the top vertex, which height is defined by the distance of the two touches. To select an object the spherical cursor has to intersect the object. After completing the selection the object is attached to the spherical cursor and is moved and rotated with it until it is deselected. To manipulate virtual objects in full 9 DOF, Toucheo [18] proposes 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 (Fig. 2.11(b)). The authors combined a bi-dimensional TRS interaction on the surface with the balloon metaphor and a 3D transformation widget that provide both the remaining rotations and independent scale along three axes. This widget is composed of a central disk and virtual rods. Dual-touch gestures



(a) Balloon Selection technique [17].

(b) Toucheo System [18].

Figure 2.11: Manipulations on stereoscopic tabletops.

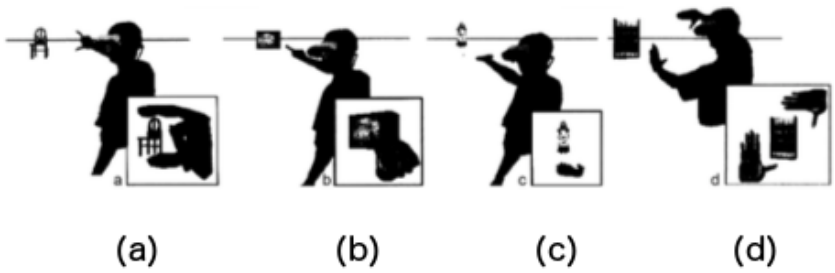


Figure 2.12: Image plane interaction techniques [20]: (a) Head crusher, (b) Sticky fingers, (c) Lifting palm, (d) Framing hands.

on the central disk result in TRS operations on a plane parallel to the touchscreen (XY Plane). X and Y rotations are triggered by finger crossing the selected virtual rod, allowing the control of a single DOF. Scaling widgets allow the control of the X and Y scale factors while for Z translations and scaling operations, the authors used an approach inspired from the balloon metaphor [17]. The user first touches the center of the disk with one finger, and then adjusts the height and the size with a second finger.

2.4 Mid-Air Manipulation

Having an input with higher DOF, most current mid-air approaches for 3D virtual object manipulation try to mimic physical world interactions. However, fine-grained manipulation and precision tasks are hard to perform with these techniques, due to limited human accuracy, which is sometimes aggravated by input devices' resolution.

Pierce et al. [20] believed that the use of 2D image plane for interaction in 3D scenes extends beyond desktop interfaces, traditional mouse/keyboard applications, to IVE. They developed a number of selection techniques for objects at a distance and a set of manipulation techniques as well (Fig. 2.12). For instance if the user wishes to perform a close-in operation on a distant object, the object must be translated in a non-jarring, sensible manner. This involves object scaling, by resizing the object automatically to fill a convenient working volume or dynamically depending on the distance from the user. The authors also considered the option to leave the object small and allow the user to scale it explicitly. In stereo

Figure 2.13: Voodoo Dolls technique [21].

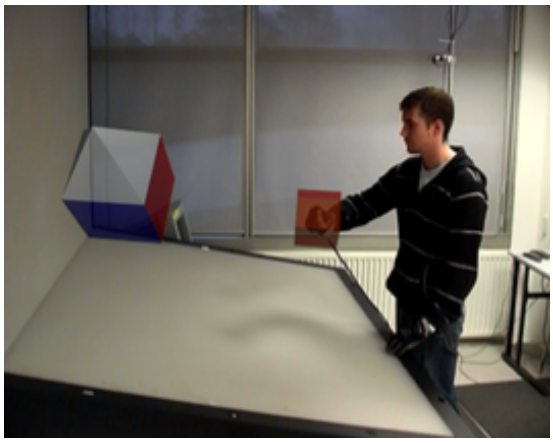


Figure 2.14: IR Technique [22].

systems such solutions face a few problems. The first problem is choosing the left or right eye's image plane, because finger position is different in the 2D image for each eye, this can be an issue if different objects are present in each side. Arm fatigue is another problem since the user keeps working with his arms extended. Object occlusion can also occur if the object is small or far away. A couple of years later Pierce et al. [21] proposed a different solution completely independent of the object scale. The Voodoo Dolls technique (Fig. 2.13), dynamically creates dolls that are hand held copies of the objects they represent. 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. Thanks to this technique the user can work at multiple scales without explicitly resizing objects, avoiding some of the problems described above.

We have seen the influence of degrees of freedom and their separation on a multi-touch environment. This separation provides a predictable and efficient way to manipulate objects even though the user performs one operation at a time. Veit et al. [22] studied this influence for mid-air orientation tasks, resulting in the creation of two techniques, IR (Fig. 2.14) and BCPR (Fig. 2.15). The first technique allows users to manipulate a remote object, by grabbing a virtual manipulator with their dominant hand. This virtual manipulator, a cube, affects directly the object being manipulated, in such a way that if the

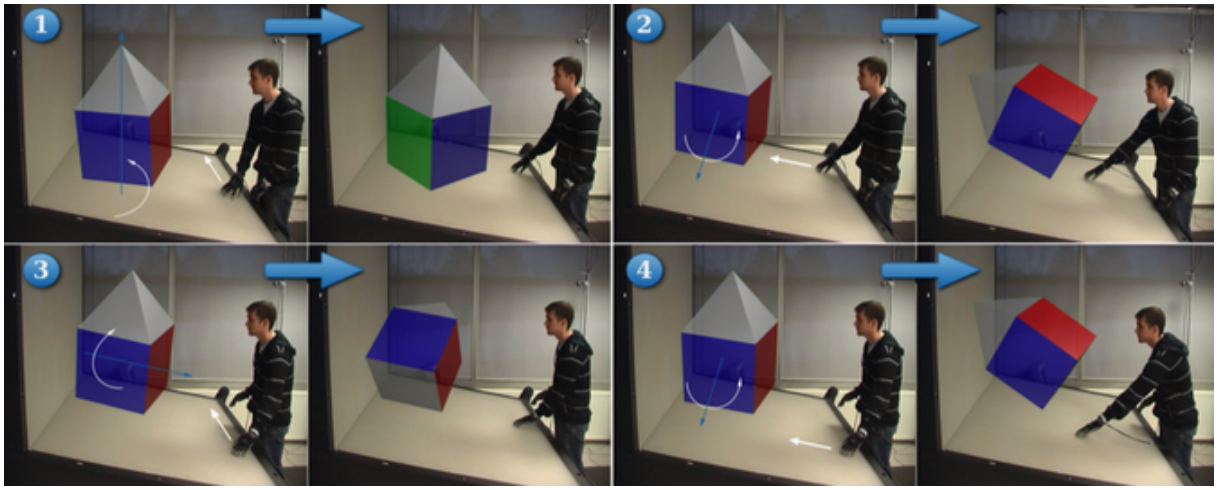


Figure 2.15: BCPR Technique [22].

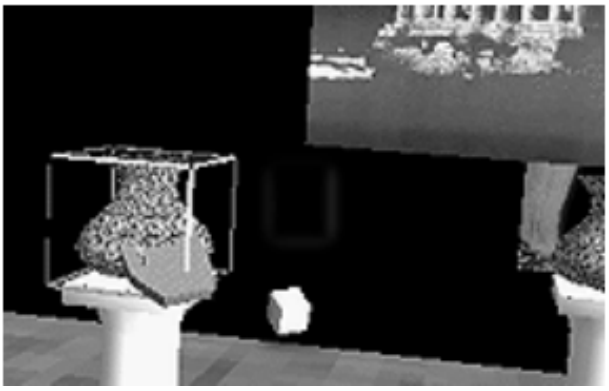


Figure 2.16: Go-Go technique [23].

user rotates the cube, the object will also rotate. Using the IR technique, users are able to combine three axes of rotation into a single gesture. On the other hand BCPR, simulates the use of a touch screen by requiring users to touch a surface with their forefingers to realize a rotation. The scene is still displayed in mid-air but since there's no virtual manipulator anymore, the user doesn't need to switch attention between the manipulated object and the manipulation widget. Rotations are performed by moving the dominant hand along the corresponding screen axis (2 axis) while the non-dominant hand gives access to the third axis. Using the BCPR technique, users were able to decompose one 3DOF orientation task into three 1DOF sub-tasks by manipulating one axis at a time. The level of precision reached by the users both in IR and BCPR techniques proves once again that separating DOF is a viable alternative for object manipulation even for mid-air solutions.

Poupyrev et al. [23] proposed the Go-Go technique several years ago. This technique expands the user's reach by changing the virtual arm's length at will, providing a natural way to grasp and manipulate objects located both far away and close to the user. Although this technique doesn't provide a precise object manipulation, control/display (C-D) ratio variations similar to this one were applied in more recent work. To overcome the lack of precision with object positioning techniques in IVEs, Frees et al. [25] proposed PRISM (Precise and Rapid Interaction through Scaled Manipulation). This technique scales the hand movement down to increase precision. Switching between precise and direct mode occurs according to the current velocity of the user's hand (Fig. 2.17). When moving an object from one general place to another, the user is not necessarily interested in being precise and moves relatively rapidly. When users are focused on accurately moving an object to very specific locations, they normally slow their hand movements down 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, reducing the effect

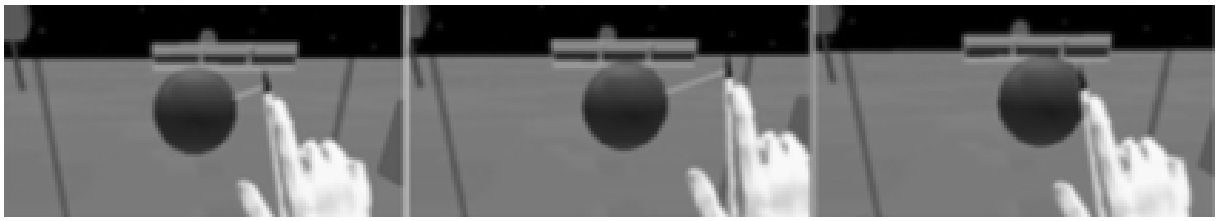


Figure 2.17: PRISM technique [24]: Offset accumulation.

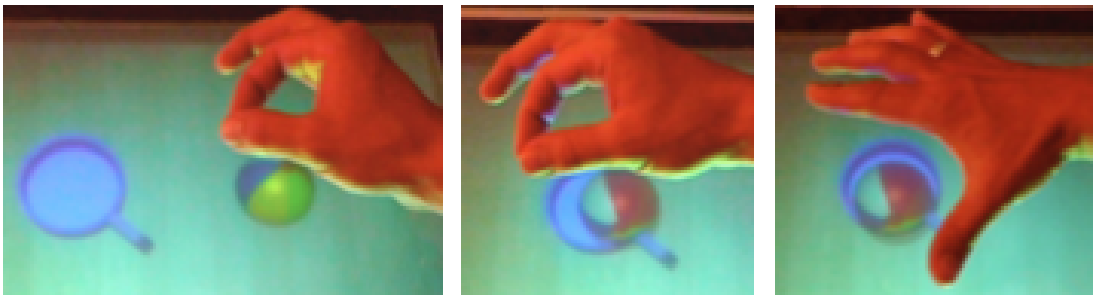


Figure 2.18: Interactions in the air [30]: Object pinching.

of hand instability and creating an offset between the object and the hand. User evaluation's results show faster performance and higher user preference for PRISM over a traditional direct approach for translation tasks. The authors later extended their previous work, by adding support in PRISM for object rotation, which uses the angular speed of the hand [24]. Although extending transformations to additional 3 DOF (all of them for rotations), authors concluded that this approach for rotations is confusing to users, while the explicit offset between screen-space and motor-space movement affects negatively interaction ease of use [26].

Also focusing on precise positioning of 3D virtual objects in IVEs, Osawa [27] proposed a position adjustment that consists in a scale factor for slowing hand movement, similar to PRISM, and viewpoint adjustment, that automatically approaches the viewpoint to the grabbed point so that the object being manipulated appears larger. Through a user evaluation, the position and viewpoint adjustment methods showed improvements for small targets over a base scenario where this adjustments were disabled.

Following both Go-Go and PRISM proposals, Chris Auteri et al. [28], decided to blend both techniques to increase precision for extended reach 3D manipulation. The solution starts by applying PRISM to the physical hand (Base Cursor) movement directly, which calculates a new cursor position (Prism Cursor) based on velocity-based scaling. Then, the distance the Base Cursor moved is amplified by the Go-Go distance-based heuristic to provide Virtual Cursor to reflect the combination of PRISM and Go-Go. The combination of Go-Go and PRISM brought a number of improvements, namely task completion success and fine-grained manipulation. Konig et al. [26] also contributed to this approach with the Adaptive Pointing technique which dynamically adjusts the C-D gain depending on the movement velocity and the current offset between the motor-space position and screen-space position. As soon as a predefined minimal velocity threshold is met the C-D gain is smoothly decreased providing a better solution than PRISM, since it simulates absolute pointing behaviour. Gallo et al. [29] also developed a technique that automatically filters and adjusts noise and hand tremor. The user is required to initially execute a dwelling task, namely to try to keep the cursor steady on a target visualized on the screen, then the logged data is used to configure the automatic filter.

Otmar Hilliges et al. [30] developed a rear projection-vision system that uses a switchable diffuser to extend the input space for interaction beyond the tabletop. This way users can interact with objects (pinching) in the scene using translation only interaction and multi-touch input to apply friction and collision forces to move them in 2D (Fig. 2.18). Additionally users can gesture directly above a virtual object

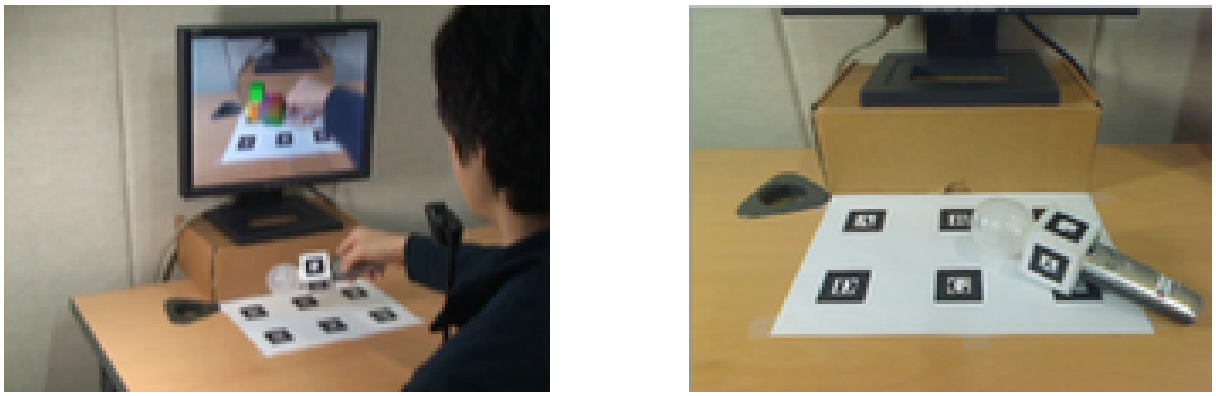


Figure 2.19: Virtual hand techniques [31]: Environment setup.

in the 3D scene, which allows in-the-air interactions with the objects. The authors also begun to explore another tabletop configuration that emulates grasping instead of pinching but they concluded that both systems have their own strengths and weaknesses. Despite being interesting ideas, both configurations have fidelity issues, tracking errors and limited DOF, making precision based tasks harder to carry out by users.

Taejin Ha and Woontack Woo [31], presented an empirical evaluation of virtual hand techniques (Fig. 2.19). Each technique has a different prop in order to meet different purposes. The first technique, named CUP, was used for 3D object selection and translation/rotation manipulations based on a 2D plane. A cylinder-shaped cup prop together with a picking and dropping gesture, allowed the user to accomplish a selection condition usually defined by how close the object was to the prop's boundaries. The PADDLE manipulation technique is similar but instead of a cylinder-shaped cup users have access to a paddle, with the object selection and manipulation being performed by picking and tilting gestures. The authors also evaluated an enhanced version of the PADDLE, called ExPADDLE, which relies on the same principle, but features, a virtual spherical selection region, a multiple tracking pattern printed box, a button for discrete input and a mouse handle for grasping it. Lastly the CUBE technique, where the user, in order to select the 3D object, must create and maintain a collision state for a specific period of time between the desired object and a virtual line tip augmented on the front of the cubical prop.

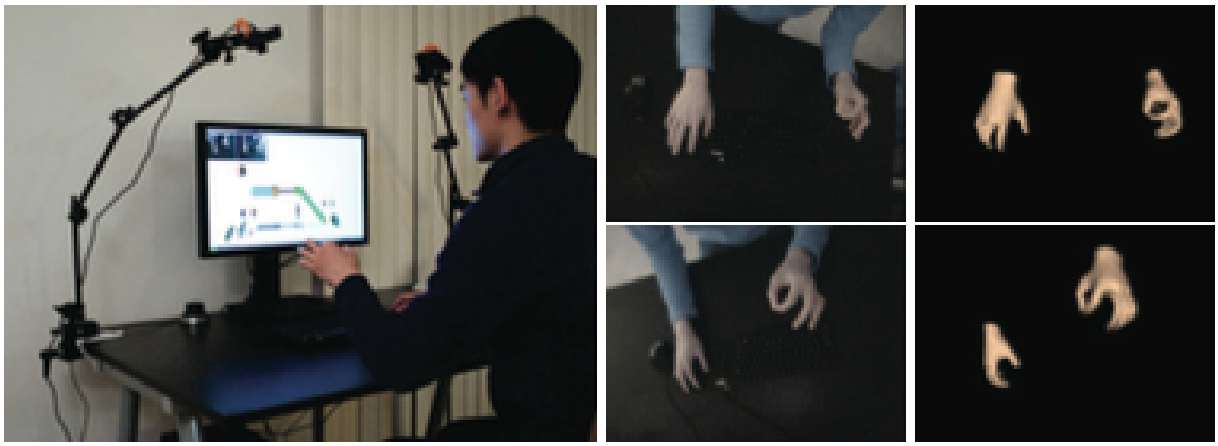


Figure 2.20: 6D Hands technique [3]: Experimental setup, camera inputs and pose estimates.

The use of props such as the CUP and PADDLE seemed easy to understand as noted by the user tests carried out by the authors. Some participants actually found similarities with physical world tasks, making object manipulation predictable and easy to perform. It is also important to note that the discrete input included in ExPADDLE, affected positively the selection/release task, due to its similarities with the classical mouse/keyboard input devices.

We have seen different approaches to mid-air manipulation. Some try to tackle it by adjusting control/display (C-D) ratios dynamically while others recur to metaphors and physical props in order to increase predictability and user intuition. Robert Wang et al. [3], presented a markerless hand tracking system, designed for CAD applications (Fig. 2.20) with a focus on user comfort. The technique called 6D Hands, offers 6 degrees of freedom for each hand and a pinching gesture for selection. To recognize these gestures, the authors built a data-driven pose-estimation system, with the data being gathered by a camera setup and stored in a sampled database. This way hand tracking is performed without any markers and it is even possible for the user to rest his elbows while performing tasks and work alongside the mouse and keyboard. However complex operations like free object placement are not possible since snapping is automatically executed when objects are close to each other. Keyboard and mouse are still necessary for other tasks such as entry of numerical values, annotations, or menu navigation.

Nguyen et al. [32] proposed a widget consisting of four manipulation points attached to objects, called 3-Point++ tool, which include three handle points and their barycenter (Fig. 2.21). The barycenter can be used for approximate positioning through 6 DOF. If one handle point is manipulated, the object is rotated around an axis created by the two other handle points. If two handle points are manipulated at the same time, the object is rotated around the third handle point. An evaluation was carried out comparing the 3-Point++ tool with a direct 6 DOF approach. The 3-Point++ technique had worst results due to its complexity. Extending the previous work, Nguyen et al. [33] presented the 7-Handle manipulation technique. This technique consists of a triangle shaped widget with seven points as depicted in Figure 2.22. Three points called first-level handles, are the three vertexes 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

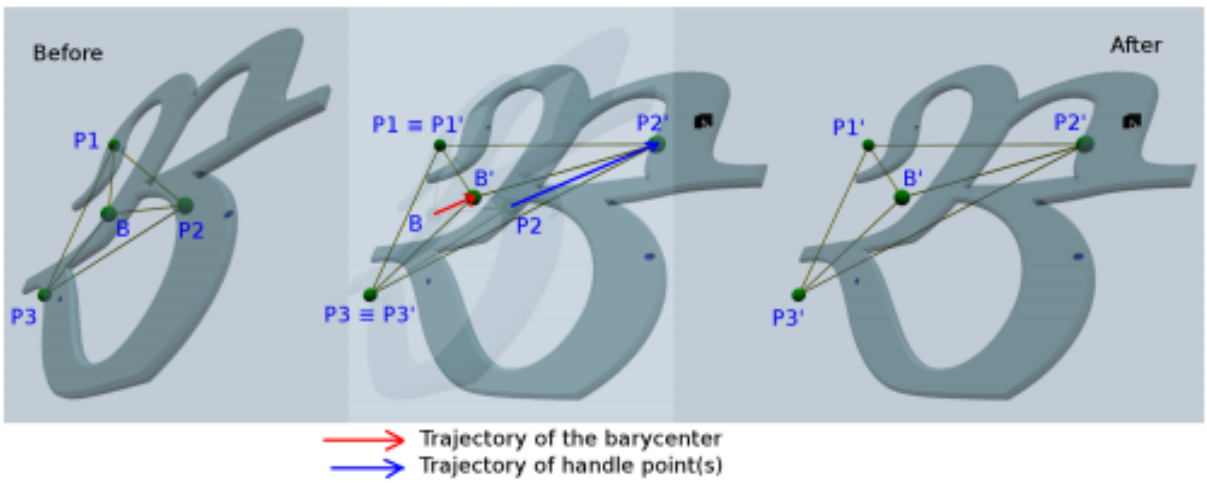


Figure 2.21: 3-Point++ technique [32]: The handle point P2 is manipulated, the object is rotated around an axis created by the two handle points P1 and P3.

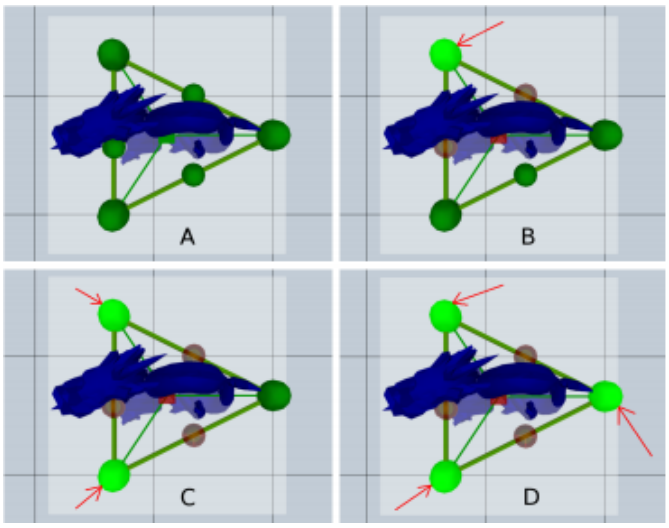


Figure 2.22: 7-Handle technique [33]: (A) No handle is controlled, (B) One first-level handle is controlled, (C) Two first-level handles are controlled, (D) Three first-level handles are controlled.

used to control its two adjacent first-level handles. The last point, the third-level handle is positioned at the centroid of the three first-level handles and can be used as a direct manipulation tool with 6 DOF. 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.

Mendes et al. [35] implemented and studied four mid-air techniques, in order to understand which one suits best for object interaction above a stereoscopic tabletop surface. 6DOF Hand allows the user to grab the object directly with one hand, drag the object in space and control object rotation with wrist rotation. Scaling is performed by varying the distance between the hand grabbing the object, usually the dominant hand, and the non-dominant hand grabbing somewhere in space outside the bounding box. 3DOF Hand works in a similar fashion, but translation and rotation are separated to prevent unwanted manipulations. Rotation is achieved by rotating the free hand wrist while keeping the object selected with the other hand. In the third technique, called Handle-Bar [34], object manipulation relies on the middle point of each hand, after the grab. Object translation is performed by moving both hands in the same direction while rotation is triggered by moving hands in different directions (Fig. 2.23). In the

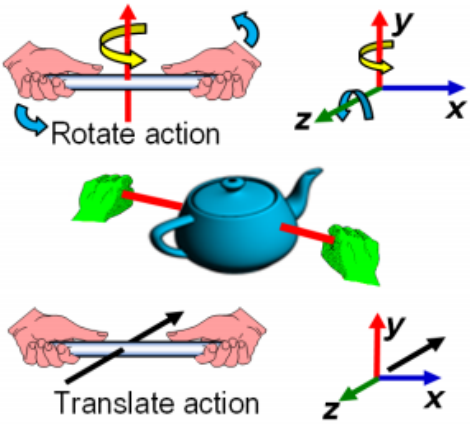


Figure 2.23: Handle-Bar technique [34].

remainder technique, Air TRS, the hand responsible for grabbing the object moves it, while the other hand, performs object rotation and scale. According to the participants, the 6DOF Hand approach was more natural to use, thanks to the direct interaction with objects. The Handle-Bar solution was as fast as the 6DOF Hand and was not affected by unwanted occlusions with the dominant hand. The authors concluded that mid-air manipulations that have a greater resemblance to interactions in the physical world appeal more to users.

2.5 Discussion

We will discuss both the advantages and disadvantages from the techniques we considered relevant to our work. In order to compare important features from different techniques, we provide a technique classification in Table 2.1. The first two columns describe the perceived space, which can be bi-dimensional (2D) or three-dimensional (3D) and represents the space apparent to the user, followed by the input space which is similar to the previous column but depicts the input space. Translation, Rotation and Scale capabilities belong to the fifth, sixth and seventh column in our table. The following aspects are translation/rotation (T/R) separation, single DOF Translation and single DOF Rotation where we consider the decomposing number of DOFs necessary to its completion as well as Translation and Rotation clear separation. Widgets column, as the name reflects, indicates if the technique in question makes use of auxiliary graphical control elements while improved precision is related to techniques that focus on object manipulation precision improvements over direct manipulation.

Multi-touch surfaces started with a more direct approach, that allows users to directly interact with the object along the surface. Sticky Fingers [8], expanded the number of available DOF in both translation and rotation by giving the ability to use multiple touch-point during manipulation. This resemblance leads to a natural and familiar interaction to users, despite the lack of DOF separation. tBox [13], GimbalBox [15] and the Balloon selection technique [17], approached this matter through virtual widgets and hand gestures. In the first two techniques users are able to control translation, rotation and scale independently. As a result error prone situations during selection and manipulation are greatly reduced. However the bi-dimensional nature of the perceived space and the input space, and the reduced surface screen size limits user interaction. Balloon selection decomposed the available DOFs into smaller sets, by making use of hand gestures. While the principle is similar to the widget-based techniques, the perceived space in this environment is three-dimensional which gives a better perception to users of the transformations being applied to the object.

Mid-air manipulation techniques are usually not restricted by the screen-space limitations we mentioned before. Voodoo Dolls [21] and Go-Go [23] allowed users to work at multiple scales, avoiding possible occlusions related to small objects in the image plane. While it presented an interesting solution, the lack of DOF separation and explicit object control made user interaction prone to error. Frees et al. [25, 24], Auteri et al. [28] and Konig et al. [26] tackled object manipulation precision using different means. These techniques scaled input down dynamically to increase precision in situations that required fine-grained adjustments. As this switch between direct and indirect manipulation is not controlled by

	Perceived Space		Input Space		Supported Transformations			T/R Separation	Single DOF Translation	Single DOF Rotation	Widgets	Increased Precision
	2D	**3D**	**2D**	**3D**	**Translation**	**Rotation**	**Scale**					
Sticky Fingers [8]	√		√		√	√						
tBox [13]	√		√		√	√	√	√	√	√	√	
GimbalBox [15]	√		√		√	√		√	√	√	√	
Balloon Selection [17]		√	√		√	√		√				
Voodoo Dolls [21]		√		√	√	√						
Go-Go [23]		√		√		√						
PRISM [24]		√		√	√	√						√
Go-Go + PRISM [28]		√		√	√	√						√
Adaptive Pointing [26]	√		√		√	√						√
6D Hands [3]	√			√	√	√						
6DOF Hand [35]		√		√	√	√	√					
3DOF Hand [35]		√		√	√	√	√	√				
Handle-Bar [34]		√		√	√	√	√					
Air-TRS [35]		√		√	√	√	√					
3-Point++ [32]		√		√	√	√					√	
7-Handle [33]		√		√	√	√				√	√	

Table 2.1: Techniques classification.

users, interacting with objects becomes unfamiliar and confusing. 6D Hands [3] approached the matter in a distinct way. While the input space was three-dimensional, the perceived space was bi-dimensional. This means that any manipulation performed on an object, even being in mid-air, was constrained by the screen where those transformations were shown. It is important to notice that this system focused on marker less hand tracking and was designed to work alongside a keyboard and mouse. As such these limitations were probably reasonable for the objectives set by the authors.

Mendes et al. [12] compared different mid-air techniques above stereoscopic tabletops. 6DOF Hand, 3DOF Hand and Air-TRS allowed a direct object control, but translation and rotation were separated in the last two techniques in order to avoid unwanted manipulations. This separation benefits tasks that require some degree of precision, as users have a smaller amount of simultaneous DOFs available. Handle-Bar presented similar behaviour, but required the use of two hands to translate and rotate the desired object. This presented an advantage compared to the previous techniques as any possible occlusion resulting from grabbing the object directly is avoided. Nonetheless object rotation becomes more complex to users as they have to rotate both hands in different directions. Translation and rotation separation was not available in this technique, as opposed to 3DOF Hand and Air-TRS, leading to error prone situations during manipulation.

Lastly both 3-Point++ [32] and 7-Handle [33] techniques consisted in a set of three-dimensional widgets. Similarly to the approaches previously described in multi-touch environments, these virtual widgets allow users to treat translation and rotation separately. While object manipulation could benefit from these aspects, the authors noted that the widget design lead to user error. Not only that but benefits were only significant when working with considerable large objects.

In short, none of the existing techniques follows explicit DOF separation and single DOF control in mid-air, even though it has been proven that it benefits manipulation in mouse and touch environments. Regarding manipulation precision, some techniques achieved good results by scaling the object movement down but were missing DOF separation, namely independent translation and rotation control.

Chapter 3

Prototype

In order to reach our objective we developed a prototype that enables body tracking and hand input to interact with virtual environments. In this chapter we disclose the architecture and setup, followed by an hardware description.

3.1 Architecture

Our architecture is depicted in Figure 3.1 and focuses primarily on 3 modules: the interaction module, the scene module and the render module.

The dominant hand gestures are captured by an Imuduino[1], which contains motion and orientation sensors as well as reduced dimensions and weight. A mouse ring from Genius[2] is used as a discrete

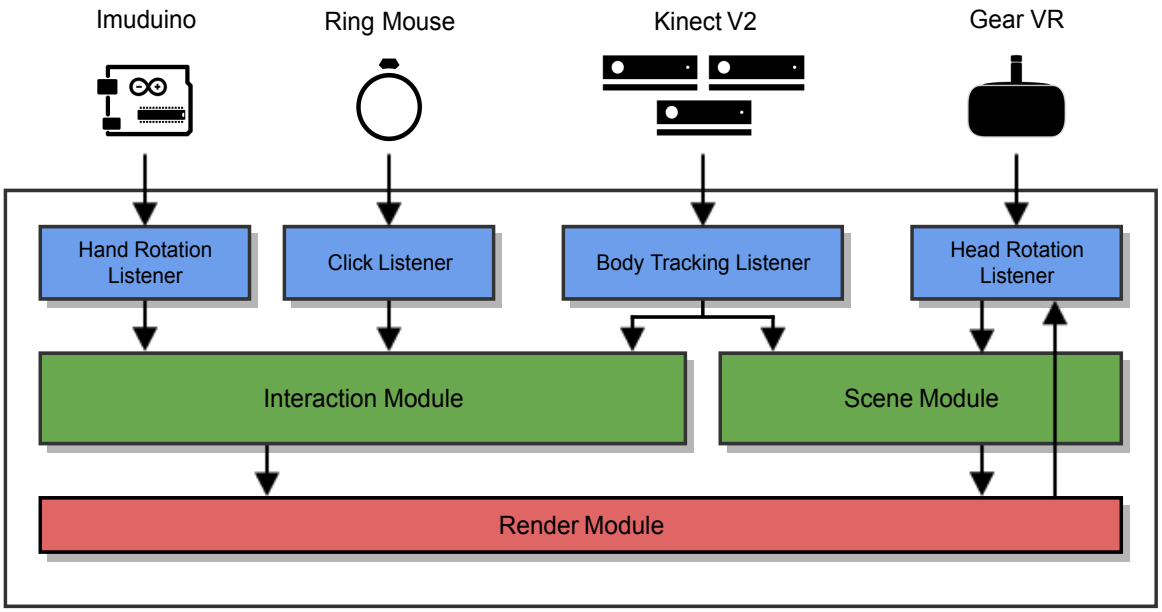


Figure 3.1: Prototype architecture.

[1] Imuduino, `http://femto.io/products/imuduino/`, last visited August 31th 2016.
[2] Genius Mouse Ring, `http://www.geniusnet.com/`, last visited September 2nd 2016.

input device in the non-dominant hand. The interaction module, where the core of our prototype resides, takes care of the data gathered by these devices and the user's hand and body position in space, captured by multiple Microsoft's Kinect V2[3]. Regarding head orientation, data collection relies on the Samsung Gear VR. This virtual reality head-mounted display (HMD), offers head tracking capabilities and a variety of development tools and documentation. The scene module contains every object represented in the scene and its placement, including the user's head orientation and body placement. Similarly to the interaction module Microsoft's Kinect V2 is used to gather user's skeleton information. Finally the render module, takes the data retrieved by each one of the previous modules, renders the scene and sends it back to the Samsung Gear VR. We used Unity 3D engine[4] with gravity and objects' collisions disabled, and C# scripting to develop our prototype and implement the techniques described in the next section. We chose this engine due to its extensive documentation, ease of use and platform compatibility.

3.2 Setup

Our prototype setup comprises non-invasive and affordable full body user tracking with three depth cameras Microsoft Kinect v2. One of them was placed facing the user while the remaining ones lied on each side, 90 degrees from the first one. Since Kinect V2 is not capable of retrieving hand orientation with a high degree of confidence, we acquire such data through an Imuduino-based custom device, pictured in Figure 3.3. 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. Object scaling was disabled during user testing.

Figure 3.2: Microsoft's Kinect V2, motion sensor.

[3]Kinect V2, `https://en.wikipedia.org/wiki/Kinect_for_Xbox_One/`, last visited August 31th 2016.
[4]Unity 3D, `https://unity3d.com/`, last visited August 31th 2016.

3.3 Body Positional Tracking

Kinect V2 is responsible for tracking users body (Fig. 3.2). Our setup is comprised by three devices as mentioned before. Each Kinect features a camera with a color image resolution of 1920 x 1080 pixels and a fov of 84.1 x 53.8 resulting in an average of about 22 x 20 pixels per degree. The environment is read through infrared using a time-of-flight camera, with a depth image resolution of 512 x 424 pixels with a FOV (Field of View) of 70.6 x 60 degrees resulting in an average of about 7 x 7 pixels per degree. Each sensor is connected to a PC running a data client. These clients are responsible for sending their data to a server, where they are further processed and broadcasted through our private network.

3.4 Hand Rotation Tracking

We used a custom made device based on an IMUduino. This device incorporates an IMU and a Bluetooth LE module (Fig. 3.3). The IMU is composed of gyroscope, an accelerometer and a digital compass sensors for accurate 3 DOF orientation tracking. This device also contains a pressure pad, which is used to switch between hand states and detect different levels of pressure. Our server receives the rotation data via Bluetooth as a quaternion which is then sent through Wi-Fi to our prototype.

3.5 Auxiliary Hand Input

Ring mouse, as the name suggests works as a traditional mouse peripheral (Fig. 3.4). While Imuduino is used in the dominant hand the ring mouse serves as a discrete input device for the non-dominant hand. This way users can interact with our prototype and make use of its capabilities to perform tasks that require both hands. Communication from the device to our server is made through a 2.4GHz RF while Wi-Fi is used to send data from the server to the smartphone where our prototype is running.

Figure 3.3: Our custom made device for tracking hand's rotation and its open / grab state.

3.6 Head Rotation Tracking

We use a Gear VR Head Mounted Display (HMD) developed by Samsung, in collaboration with Oculus (Fig. 3.5). Its use relies on a compatible Samsung smartphone that acts as the display and processing unit. The headset features a ninety six degree field of view, as well as an IMU for rotational head tracking. For discrete input a touchpad and back button is included on the side. We use this device to display our virtual environment and rotate the camera accordingly.

3.7 Sensor Fusion

Our prototype receives data from multiple sources via Wi-Fi. These sources provide discrete input and tracking data. We implemented a body tracking listener in order to receive the data sent by the server and share it with both the interaction module and the scene module. The interaction module uses the body info for interaction purposes, such as movement and distance vectors regarding users hand and body position change. It also features an adaptive double exponential smoothing filter[5] to reduce tracking noise and improve user precision at the expense of increased latency. We tried to find a good compromise in order to tackle some of the hardware tracking limitations.

Users rely on a virtual avatar to perform the required tasks. As such the scene module applies the data directly to our character model. This is achieved by making use of the positions retrieved by the three Kinects in order to calculate the avatar joint angles and place them in the correct position.

Hand rotation is complemented by Imuduino and calibrated when the user's arm is horizontal using the data provided by the depth cameras, to avoid errors. Any pressure made on the device included pressure pad is also captured and used to represent a change in the hand state, open or closed. While Imuduino is placed on users dominant hand, a different device is used in the remaining one as we did not

Figure 3.4: Genius' Ring Mouse for discrete input.

[5]Skeletal Joint Smoothing White Paper: `http://msdn.microsoft.com/en-us/library/jj131429.aspx`, last visited 17th October 2016.

Figure 3.5: Samsung's Gear VR.

have access to an extra Arduino board. Ring mouse is a small piece of hardware that provides discrete input and acts as a traditional mouse when connected to our server. Any input received by both devices is sent to the interaction module. This way users can perform tasks that rely on one or two hands.

Regarding visualization and head rotation Gear VR provides both. The interaction module and scene module are connected to a render module, which sends the visual information back to the head mounted display. Head rotation relies on the headset IMU and its functionality is exposed directly through Unity.

Users can also perform a calibration at any time, by lifting their left arm above the head and touching with the right hand in the Gear VR touchpad. This calibration resets the head rotation controlled camera and the initial body position and orientation in the virtual environment.

3.8 Summary

This chapter started by disclosing our prototype architecture and setup. Three main modules were implemented, with the interaction module being responsible for any tracking data sent by the different pieces of hardware. In order to communicate with each device four listeners were created. These listeners communicate with both interaction module and scene module while the render module renders the scene and sends it back to the Gear VR. Regarding head and camera rotation the implementation and setup was directly provided by the head mounted display IMU and the Unity API. Hardware was also detailed including device communication.

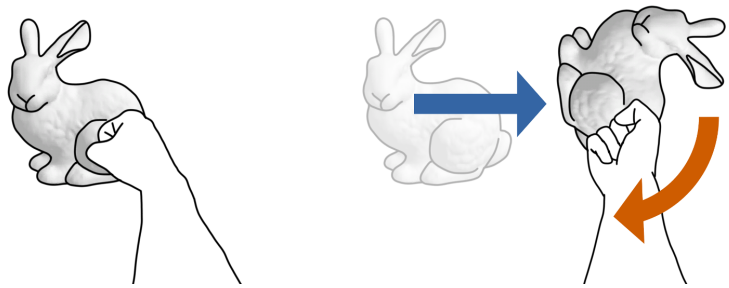
Chapter 4

Assessment of DOF Separation in Mid-Air

In this chapter we describe three mid-air object manipulation techniques used to assess the importance of DOF separation in mid-air. The first technique is a direct approach in which all transformations performed by the user's hand are directly applied to the object. As this kind of manipulation is the most natural and a common approach in mid-air it is a good baseline for our assessment. The second technique follows scaled transformations based on the user's hand speed and was designed with precision in mind making it a good approach for comparison. Lastly the third technique is our implementation of spatial widgets for separating DOF, as the use of widgets already showed benefits in mouse and touch interfaces. All techniques provide 6 DOF transformations: three for translation and three for rotation. We follow this description with a user evaluation carried out to assess the importance of DOF separation in mid-air object manipulation. Lastly we summarize this chapter by pointing out the most relevant topics previously introduced.

4.1 6DOF

To mimic interactions with physical objects as closely as possible, direct manipulation uses all 6 DOF information from users' hands [3]. It is often used as a baseline for evaluations of other techniques [24, 35, 33]. 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 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.



(a) Initial State.

(b) Simultaneous translation and 90 degrees rotation.

Figure 4.1: 6DOF technique.

4.2 PRISM

We implemented the PRISM technique as presented by Frees et al. [25]. This technique aims in improving accuracy of direct manipulation, switching between a precise and a direct mode according to the current speed of users' hands. Hand's movement in each coordinate axis is scaled down when users move their hands slower than a pre-defined threshold (SC) 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, translating into a

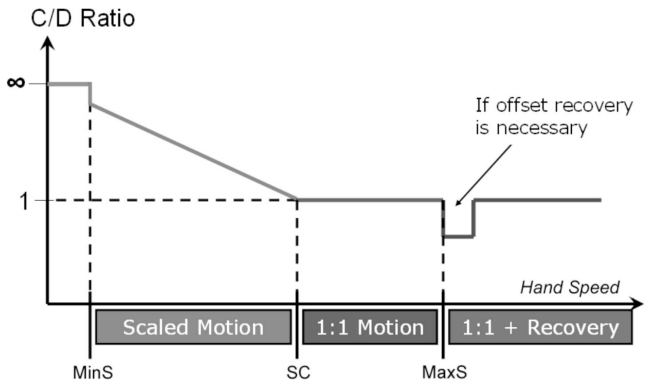
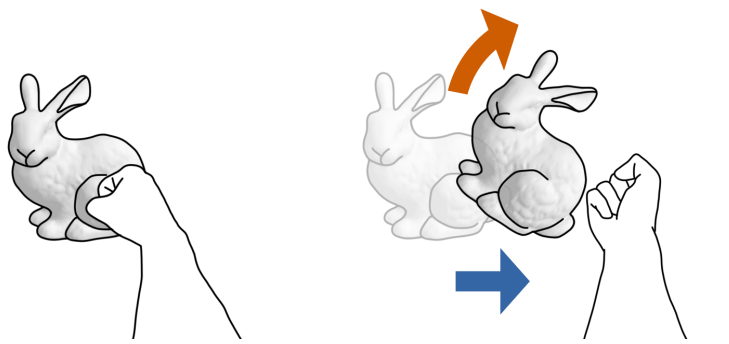


Figure 4.2: Simplified interface diagram showing how PRISM [25] uses Hand Speed to adjust CD.



(a) Initial State.

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

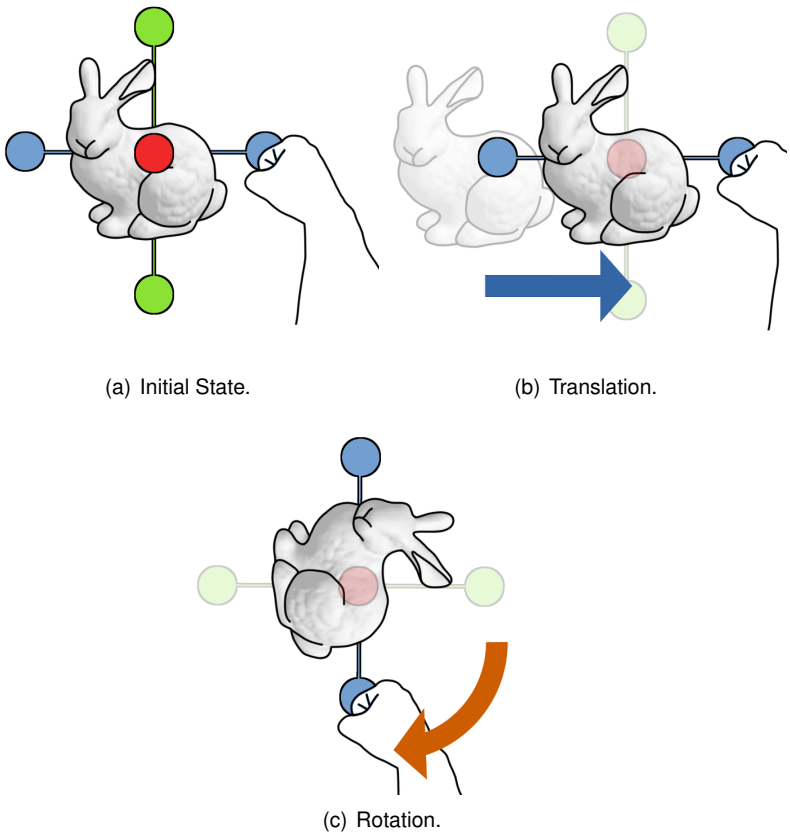
Figure 4.3: PRISM technique.

1 to 1 motion (Fig. 4.2). We also included rotations later proposed by same authors [24], which follows the same premise from translations, scaling down slow wrist rotations around each coordinate axis. As suggested by the authors, resulting offsets are represented by a white line for translations, and two sets of axis for rotations. Similarly to 6DOF technique, both translations and rotations can be performed simultaneously, as exemplified in Figure 4.3.

4.3 Widgets

Widget based manipulations are widely used in mouse and keyboard 3D user interfaces. Our implementation, as opposed to those described in sections 4.1 and 4.2, strictly follows DOF 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. [5], illustrated in Figure 4.4. Each axis is represented by a cylinder shaped object and two spheres in each end while coloring follows a RGB coding for XYZ. Users can grab one of the two spheres connected to the desired axis and move the hand along the axis to trigger object translation. For rotations, the approach is similar, but the hand movement is performed around the target axis.

The decision to either perform a translation or rotation, is made based on the hand's path after 10 cm. After this distance is covered we calculate the dot product between the hand's path vector and the



(a) Initial State.

(b) Translation.

(c) Rotation.

Figure 4.4: Widgets technique.

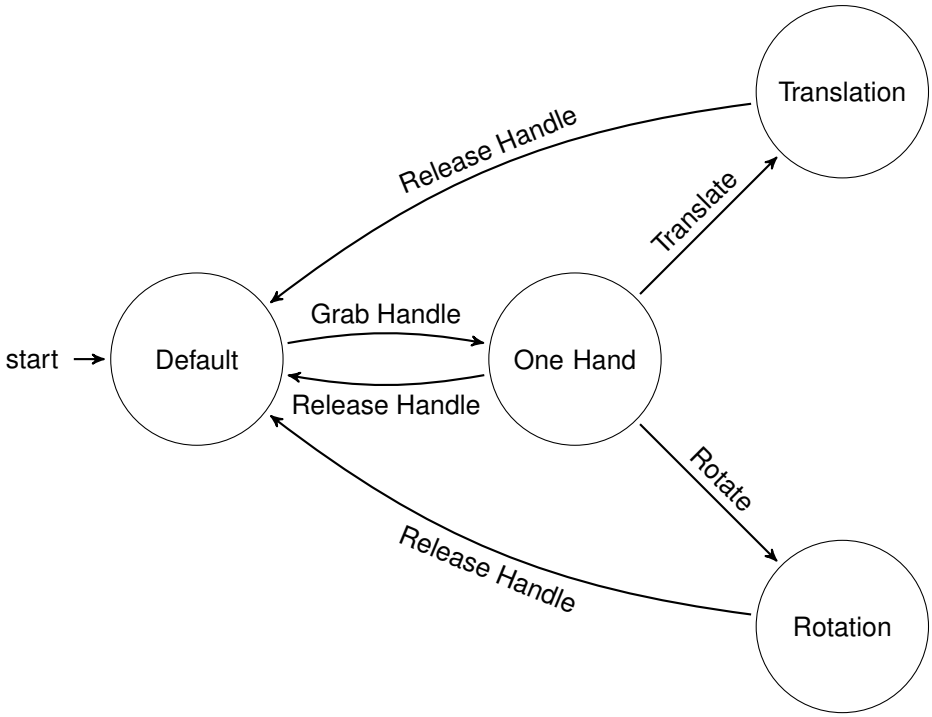


Figure 4.5: Widgets Finite State Machine.

handle's forward, up and right directions as depicted in Figure 4.6. The maximum absolute value from these results is taken and used as an estimation to reach a decision. Selected transformation and axis remain locked until a release gesture.

In order to control every possible state transition, we implemented a finite state machine (FSM). The diagram 4.5 represents the machine including all of its transitions. Two functions run every frame to determine the transition to be carried out.

Any object that can be controlled by users, starts in a state called Default State. If a user grabs an handle (sphere), we transition to One Hand State. This state represents the moment before the decision to translate or rotate the object is made. After a decision is found we transition to one of two possible states, the Translation State or the Rotation State. We stay in one of these states until a release gesture is performed, in which case we return to the Default State.

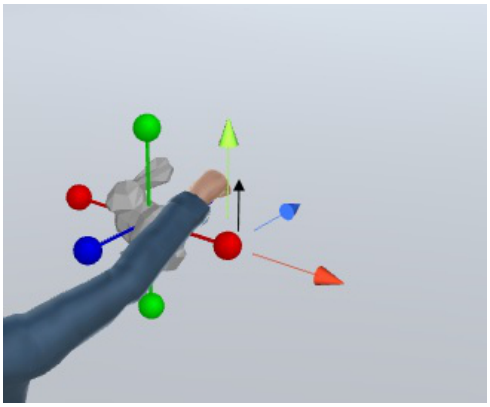


Figure 4.6: Widgets decision vectors.

4.4 Evaluation

Since DOF separation showed positive results in both mouse and touch interaction for virtual 3D object manipulation, we conducted a user evaluation to assess if it also benefits spatial interactions in IVEs. In order to do so we implemented and used the three techniques previously mentioned.

4.4.1 Methodology

All user sessions followed the same structure, each lasting approximately 45 minutes (Tab. 4.1). 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, 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 4.7. Following the practice period, participants were asked to perform a set of six tasks, described in the next section. After completing each technique's tasks, participants fulfilled a questionnaire regarding distinct aspects of the interaction. The experiment concluded with a profiling questionnaire.

4.4.2 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 [24, 11, 35], where participants had to put the exhaust pipes in the right place of a car engine[1]. That component of the engine was the only object in our virtual environment that could be grabbed and transformed. Engine's model had a semi-transparent replica of the pipes showing the only possible target position and orientation, as depicted in Figure 4.8. 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.

Stage	**Duration**
Experiment Introduction	5 min
Technique Description	10 min
Training Period	5 min
Task Execution	20 min
Questionnaire Completion	5 min
Total	45 min

Table 4.1: Methodology estimated stage duration.

[1] Original 3D model of the used engine uploaded to Sketchup's 3D Warehouse by user M-Speed.

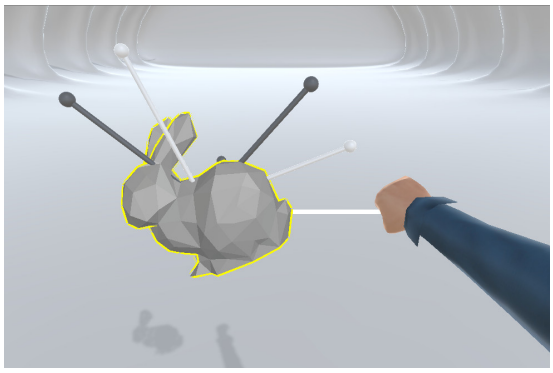
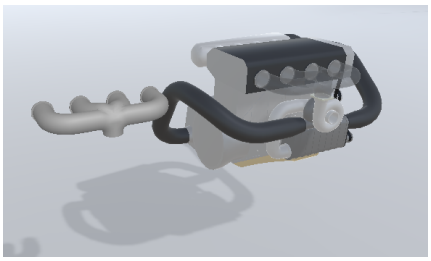
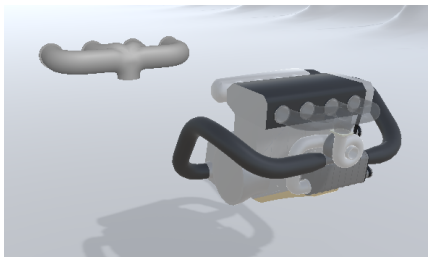


Figure 4.7: Interacting with an object in our virtual environment during the training period, with PRISM technique.

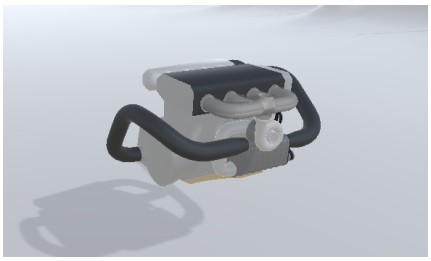
For the first task (Figure 4.8(a)), the object to be manipulated begun with the correct position along both YY and ZZ scene axes and orientation, only with an incorrect position according to the X coordinate. Similarly to the previous task, the second task (Figure 4.8(b)) object started with the correct orientation, however its position was incorrect along all three coordinates. The third task (Figure 4.8(c)) consisted in only rotating the object around the Z axis, while the fourth task (Figure 4.8(d)) implied rotation around an arbitrary axis, requiring no translation as well. The fifth task (Figure 4.8(e)) required the object



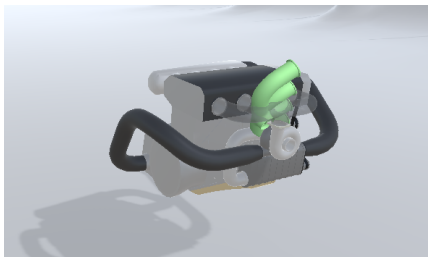
(a) Task 1.



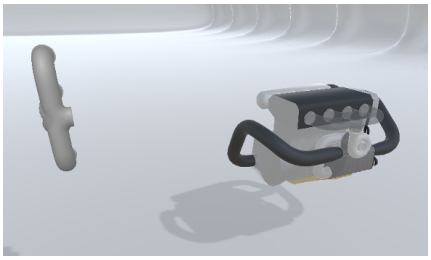
(b) Task 2.



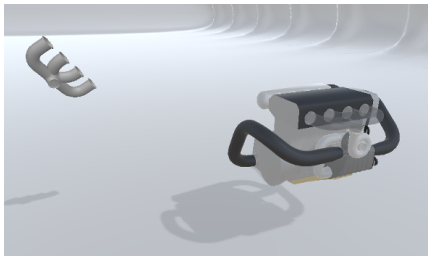
(c) Task 3.



(d) Task 4.



(e) Task 5.



(f) Task 6.

Figure 4.8: Tasks performed by the participants.

to be rotated around the Z axis and translated along both XX and YY axes. Finally, in the last task (Figure 4.8(f)), 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.

4.4.3 Apparatus and Participants

The experiment was performed in our laboratory with a controlled environment (Figure 4.9), using the setup detailed in the previous section. We counted with the participation of 21 people (5 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.

4.4.4 Results and Discussion

During our experiment, we collected both objective data, 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 with a Greenhouse-Geisser correction to find significant differences in normal distributed data, and Friedman non-parametric test with Wilcoxon Signed-Ranks post-hoc test. In both cases, post-hoc tests used Bonferroni correction (corrected sig. = sig. $\times$ 3).

Figure 4.9: Participants during evaluation sessions.

Objective Data

We measured time taken by participants to fulfil each task, as well as object placement error. Time taken for all tasks, in seconds, is depicted in the graph of Figure 4.10. Regarding errors, we registered both position error, in millimeters (Figure 4.11), and rotation error, in degrees (Figure 4.12).

For the translation only tasks, we found statistically significant differences in completion time (Task 1: χ^2(2)=25.368, p$<$.0005; Task 2: F(1.611,30.604)=9.025, p=.002). For the first task, post-hoc test revealed Widgets approach (avg=25s) to be faster than both 6DOF (avg=59s, Z=-3.542, p$<$.0005) and PRISM (avg=90s, Z=-3.823, p$<$.0005), and 6DOF to be faster than PRISM (Z=-3.267, p=.003). In the second task, PRISM (avg=102s) was significantly slower than Widgets (avg=49s, p=.008) and 6DOF (avg=71s, p=.028). For position error, differences were also found (Task 1: F(1.851,24.066)=17.474, p$<$.0005; Task 2: F(1.359,14.946)=6.653, p=.015), with Widgets (Task 1: avg=3.3mm; Task 2: avg=5.2mm) outperforming 6DOF (avg=15.0mm, p$<$.0005) in the first task and PRISM on both first (avg=10.7mm, p=.002) and second (avg=12.2mm, p=.003) tasks. The technique used also influenced rotation error (Task 1: χ^2(2)=24.500, p$<$.0005; Task 2: χ^2(2)=15.000, p=.001), with Widgets (Task 1: avg=0.0º, Task 2: avg=0.0º) achieving lower error than 6DOF (Task 1: avg=11.7º, Z=-3.724, p$<$.0005; Task 2: avg=9.8º) and PRISM (Task 1: avg=7.3º, Z=-3.408, p=.003; Task 2: 7.1º, Z=-2.803, p=.015).

Widgets might have outperformed both 6DOF 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 6DOF. We believe this occurred because transformation separation found in the Widgets technique made it impossible to take a direct path, requiring users to move in 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 (χ^2(2)=20.985, p$<$.0005), in which the use of Widgets (avg=27s) reduced time needed when compared to 6DOF (avg=53s, Z=-3.053, p=.006) and PRISM (avg=58s, Z=-3.823, p$<$.0005). For both tasks, position error revealed significant differences according to the technique used (Task 3: χ^2(2)=16.545, p$<$.0005; Task 4: F(1.619,14.575)=6.586, p=.012). Widgets (Task 3 avg=0.0mm, Task 4 avg=9.7mm) led to better positioning than 6DOF in both tasks (Task 3: avg=13.3mm, Z=-3.296, p=.003; Task 4: avg=15.7mm, p=.008) and than PRISM in the third task (avg=16.6mm, Z=-3.059, p=.006). Rotation error was also significantly affected by the techniques (Task 3: χ^2(2)=20.118, p$<$.0005, Task 4: χ^2(2)=16.545, p$<$.0005). Once again, Widgets (Task 3 avg=1.8º, Task 4 avg=5.3º) performed better than 6DOF in both tasks (Task 3: avg=8.8º, Z=-3.547, p$<$.0005; Task 4: avg=8.7º, Z=-2.868, p=.012) and than PRISM in the third task (avg=7.1º, Z=-3.574, p$<$.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

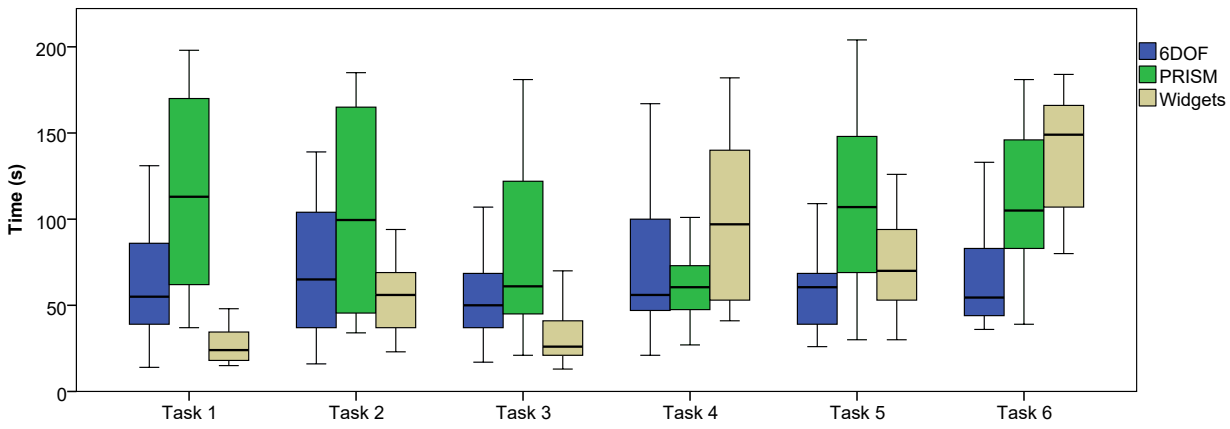


Figure 4.10: Time to complete the six tasks using the three techniques, in seconds. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

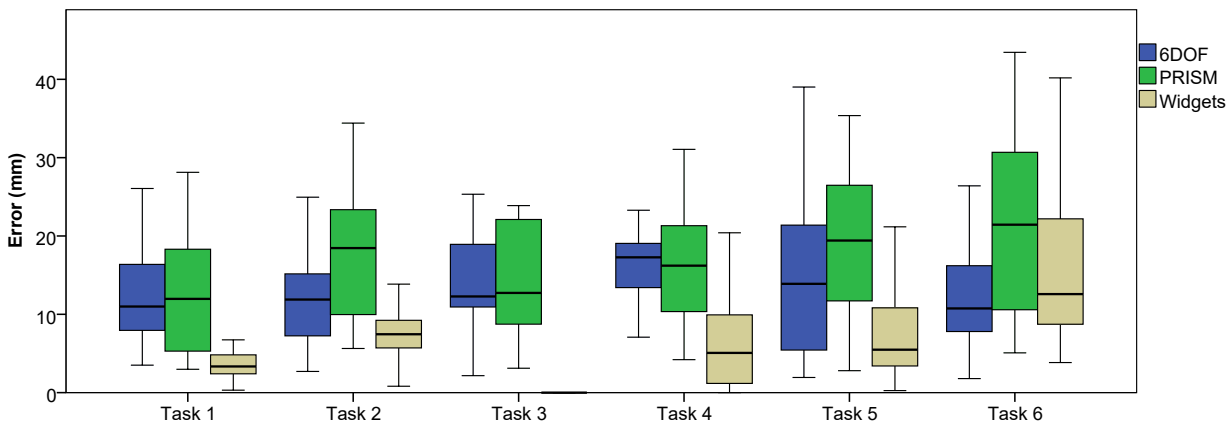


Figure 4.11: Position error attained in the six tasks using the three techniques, in millimeters. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

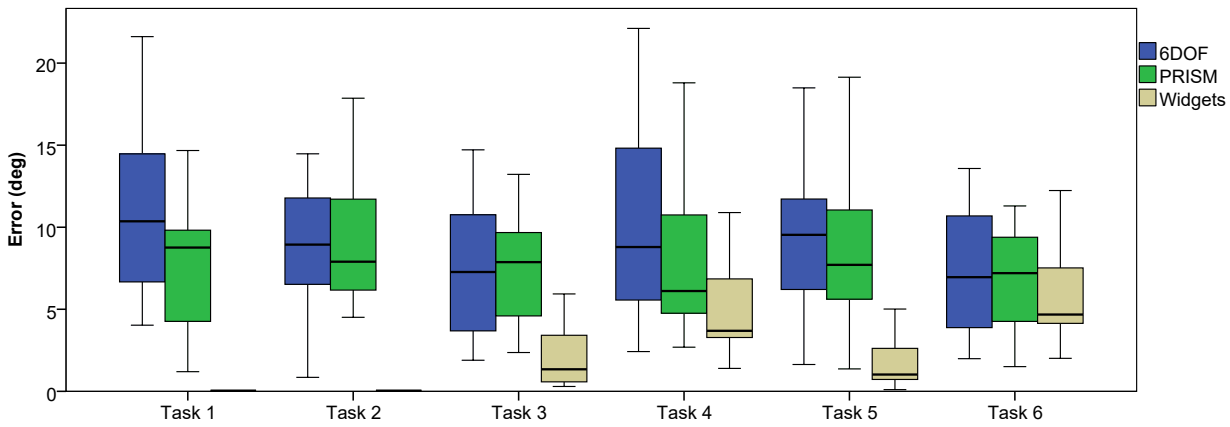


Figure 4.12: Rotation error attained in the six tasks using the three techniques, in degrees. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

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$<$.0005; Task 6: χ^2(2)=27.900, p$<$.0005). While in the fifth task PRISM (avg=102s) was outperformed by both Widgets (avg=72s, p=.004) and 6DOF (avg=63s, p=.003), in the sixth Widgets (avg=135s) took longer than 6DOF (avg=55s, Z=-3.920, p$<$.0005) and PRISM (avg=112s, Z=-2.520, p=.036). 6DOF was also faster than PRISM in the final task (Z=-3.323, p=.003). In both tasks, there were differences regarding error in object positioning (Task 5: χ^2(2)=8.533, p=.014, Task 6: F(1.671,23.391)=5.232, p=.017). Widgets (avg=6.6mm) reduced distance to target in the fifth task when compared to 6DOF (avg=15.1mm, Z=-2.809, p=.015) and PRISM (avg=21.4mm, Z=-3.010, p=.009). In the last task, 6DOF (avg=11.4mm) allowed users to place the object closer to its target position than PRISM (avg=21.2mm, p=.048). Analysing rotation error, we only found significant differences in the fifth task (χ^2(2)=22.625, p$<$.0005), in which Widgets (avg=1.1º) attained better results than 6DOF (avg=9.2º, Z=-3.823, p$<$.0005) and PRISM (avg=8.9º, Z=-3.464, p=.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 6DOF 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 or rotation error. Moreover, PRISM and 6DOF consistently shared similar results. As the authors pointed out, PRISM rotations are confusing for some users, which might have had a negative impact in tasks overall performance.

Looking at the assessment results we can draw some conclusions regarding the research questions introduced in Section 1.1. We divided user tasks in three categories, depending on the number of DOF required for completion, to provide better answers to our questions. Single DOF tasks are defined as simple tasks, three DOF tasks are moderate while complex tasks require users to use translation and rotation with three DOF each, 6DOF in total. The answers to the research questions are depicted in Table 4.2. Concerning the first research question single DOF separation does indeed reduce position and rotation error in simple tasks while in complex tasks there are no significant differences. In moderate tasks this is not verifiable with 95% confidence in all cases, but it holds true with 90% confidence

	Research Question 1	Research Question 2
Simple Tasks (1DOF)	Yes	Yes
Moderate Tasks (3DOF)	No	Yes
Complex Tasks (6DOF)	No	No

Table 4.2: Research question analysis.

	6DOF	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)

* indicates statistical significance

Table 4.3: Participants preference for each technique, regarding different criteria (Median, Inter-quartile Range).

(Task 2 Position Error (ANOVA): p=.06; Task 4 Position Error (ANOVA): p=.066; Task 4 Rotation Error (Wilcoxon): p=.06). In terms of completion time, approached in the second research question, single DOF manipulation is faster in simple tasks and slower in complex tasks. In moderate tasks Widgets were as fast as 6DOF, as we did not found any significant differences.

Subjective Data

We asked the participants how they felt about each technique using questionnaires. This included general easiness of use, translation/rotation difficulty and fun factor. Participants were given a Likert Scale from 1 to 5 to answer our questions, being 5 the favorable value. Answers are depicted in Table 4.3.

Analysing attained results, we identified significant differences in ease of use (χ^2(2)=19.547, p$<$.0005), rotation difficulty (χ^2(2)=25.352, p$<$.0005) and fun factor (χ^2(2)=13.216, p=.001). Participants strongly agreed that PRISM was generally harder (Widgets: Z=-3.716, p$<$.0005, 6DOF: Z=-3.157, p=.006) and less fun to use (Widgets: Z=-3.057, p=.006, 6DOF: Z=-2.463, p=.042). Widgets appealed more to participants to perform object rotation than 6DOF (Z=-2.863, p=.012) and PRISM (Z=-3.874, p$<$.0005). Also, participants agreed that it is easier to rotate objects using 6DOF than PRISM (Z=-2.708, p=.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 other techniques. It is as fun as direct manipulation, but with increased final placement.

4.4.5 Guidelines for Mid-air Object Manipulation

As a result of our evaluation, we were able to draw some guidelines for object manipulation in IVE. These should aid researchers and developers in creating better techniques that can combine the better aspects of each evaluated approach:

- Direct manipulation (6DOF) 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 6DOF 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. Separated scaled rotation in each coordinate axis confused participants, but they found scaled translations to be helpful in improving accuracy. Combining scaled translations with other approaches might improve their overall performance.

4.5 Summary

This chapter began with a description of three mid-air object manipulation techniques used to study the importance of DOF separation in mid-air. In the first technique any transformations carried out by the user's hand are directly applied to the object while the second scales both translation and rotation transformations depending on the user's hand movement speed. The remaining technique allows users to separate DOF by making use of spatial widgets. This study showed that DOF separation indeed benefits spatial manipulation, at the expense of increased completion times in complex tasks. We concluded this assessment by drawing some guidelines about mid-air object manipulation.

Chapter 5

Proposed Approach: WISDOM

Following the assessment results our proposal improves upon the three techniques described in the previous chapter by combining the strongest aspects of each one. WISDOM (Widgets combining Scaled movements and DOF separation for Object Manipulation) provides the direct manipulation found in 6DOF and the scaled movements we observed in PRISM, while maintaining DOF separation offered by our widget implementation.

5.1 Technique overview

Users can switch between different manipulation modes by enabling or disabling widgets at any moment. This behaviour can be achieved thanks to the pressure pad on our custom device 3.3, that we use to identify two different pressure levels. Direct object manipulation is set off by grabbing the desired object when widgets are disabled, while a similar behaviour with widgets enabled, triggers isolated scaled translation similar to PRISM [25]. Scaled movements are also applied during widget translations, allowing for fine-grained adjustments after users lock the desired axis.

An additional step to the transformation decision process was added as well. Previously users needed to cover a distance of ten centimeters after which a decision would be made based on the hand's

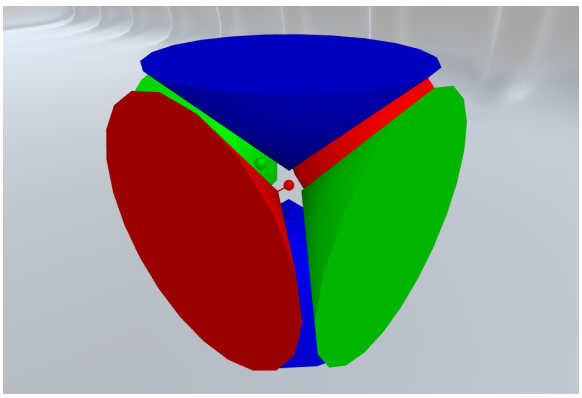


Figure 5.1: Red handle decision regions. In this case blue and green cones represent rotations around the blue and green axes respectively, while red cones represent translations along the red axis.

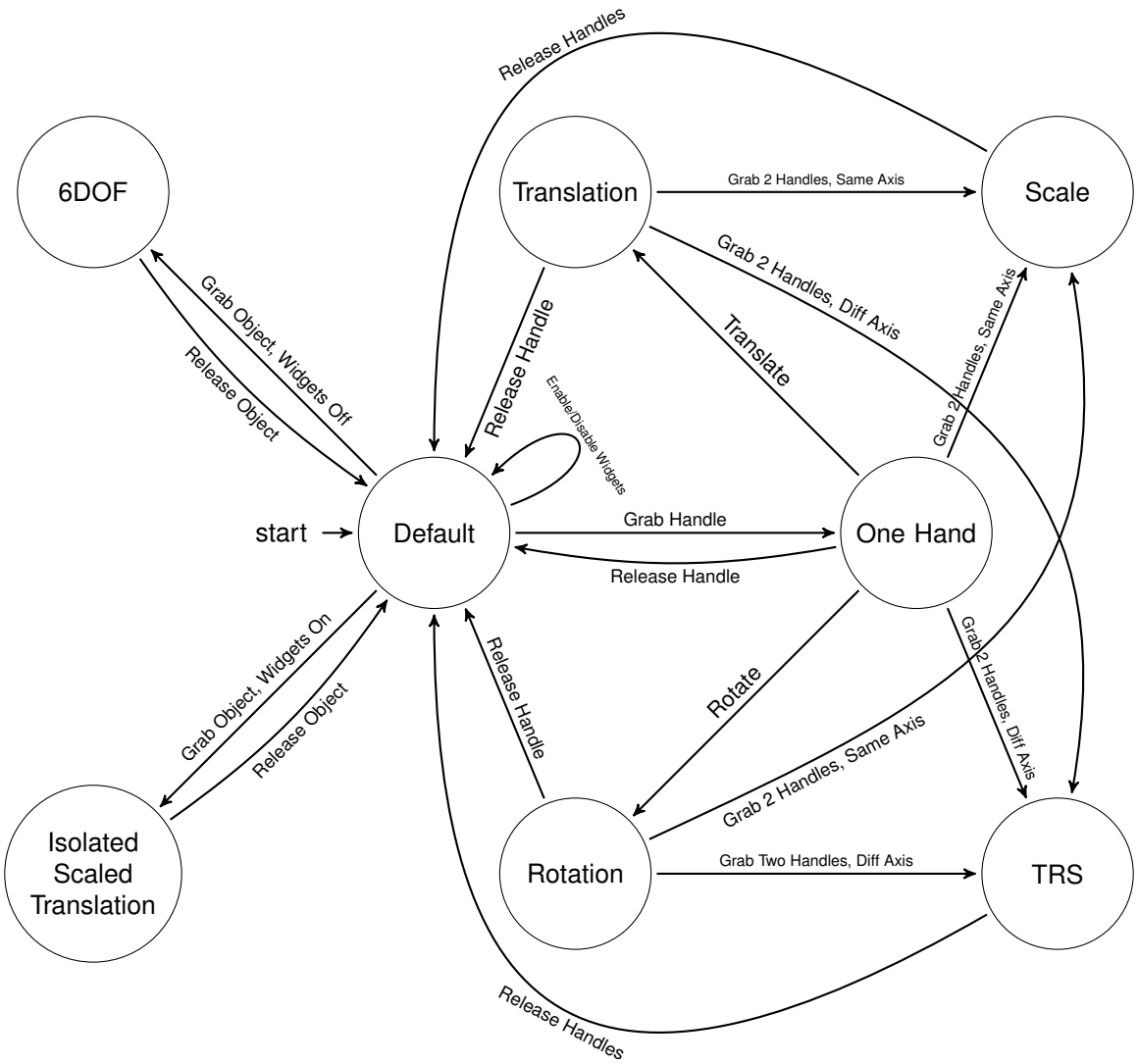


Figure 5.2: WISDOM Finite State Machine.

path. In our proposal the decision distance to either perform a translation or rotation was increased to fifteen centimeters and decision regions were added to every available handle. These regions are three-dimensional cone-shaped objects not visible to users (Fig. 5.1), that allow decisions to be made without the necessity to describe a certain hand path. In addition we implemented uniform and non-uniform object scaling, a feature not present in most mid-air techniques, and 2D TRS. Users can trigger scaling and TRS by selecting two handles from the same axis or from different axes respectively.

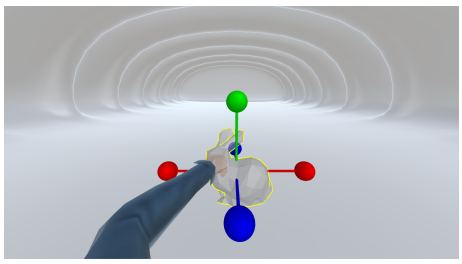
We also expanded the state machine made for our Widgets technique, with four states added in total. Direct manipulation and isolated scaled translation resulted in two states. The transition to these states is defined by grabbing the object with widgets off or on respectively. In the case two handles in the same axis or different axis are grasped we transition to Scale State or TRS State. A complete state machine representation can be found in Figure 5.2.

5.2 Direct manipulation

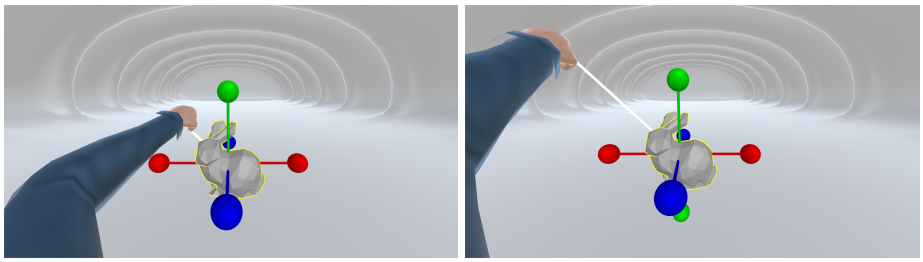
As previously mentioned in the technique overview, WISDOM features 6DOF direct manipulation. From our initial evaluation we concluded, that tasks that required users to perform a bigger number of operations, saw an increased completion time using our Widget implementation compared to the other techniques. These results together with user feedback, suggest that the combination of direct manipulation with Widgets might lead to a decrease in completion time while performing complex tasks. As a consequence we decided to approach this matter and feature both in our proposal. Users can access this mode by grabbing the object when Widgets are disabled. If for any reason fine-grained adjustments are required we can re-enable widgets at any given time by quickly reaching the second pressure level.

5.3 Widgets

Widgets presented significant advantages during user testing over the remaining techniques, but the decision to either perform a translation or rotation was susceptible to error. In order to circumvent this issue we increased the distance decision to fifteen centimeters and implemented decision regions. These regions are simple three-dimensional cone-shaped objects not visible to users, placed in the handle vicinity. Each handle has 6 regions (Fig. 5.1), representing every possible way to either rotate or translate the object through it. If users maintain their hand for at least half a second within the desired region, the operation and axis is automatically locked. By allowing users to choose the desired transformation in advance, we expect to improve decision and avoid unwanted decisions due to hand instability. The distance decision is used as an alternative if the hand covers fifteen centimeters before triggering any region. This way fast hand movements, immediately translate to a decision and apply the distance covered by the hand since grabbing the handle directly to the object.



(a) Grab object.



(b) Translate the object along an arbitrary axis.

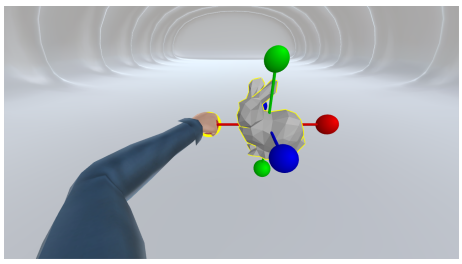
Figure 5.3: Isolated scaled translation.

5.3.1 Translation

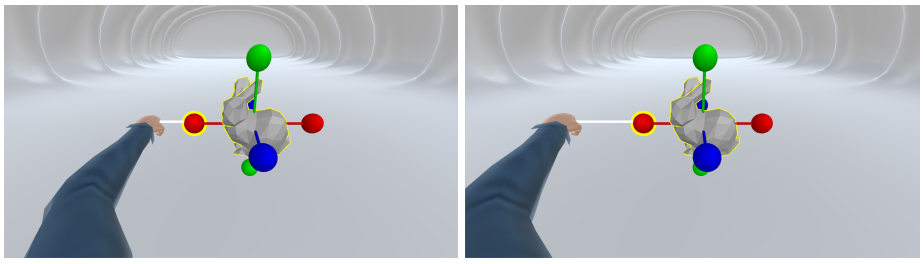
In this subsection we will approach translation outside direct manipulation. This behaviour can be achieved in one of two ways, the first one is triggered by grabbing the object directly when widgets are enabled while the second makes use of widgets to perform a translation.

In our first case users are able to translate the object in any direction with the benefits found in PRISM (Fig. 5.3). The hand speed in PRISM was determined by taking a sample of the current hand position and the hand position five hundred milliseconds in the past. The authors considered this interval to be reasonable and large enough so that small movements back and forth within the last five hundred milliseconds would be filtered out. However we found this implementation to be insufficient as noise could potentially emerge outside of the interval defined by the authors. Instead of using only two hand position samples, we capture and store the hand position and respective time stamp every frame in order to calculate the weighted average. Frame data already outside the interval is discarded and removed. We expect to achieve smoother results with this approach while maintaining it frame rate independent. The decision to leave rotation out was due to the negative impact from having both translation and rotation simultaneously available during manipulation, as well as PRISM rotation being confusing to users, as we determined in our initial evaluation.

Isolated translation was already provided in our Widgets implementation. By grabbing the desired handle and performing a certain hand path, the object would lock to translation on a single axis. We kept this behaviour but added scaled movement, to perform translations with different degrees of accuracy (Fig. 5.4). We believe that this approach might lead to improvements when fine-grained adjustments are needed.

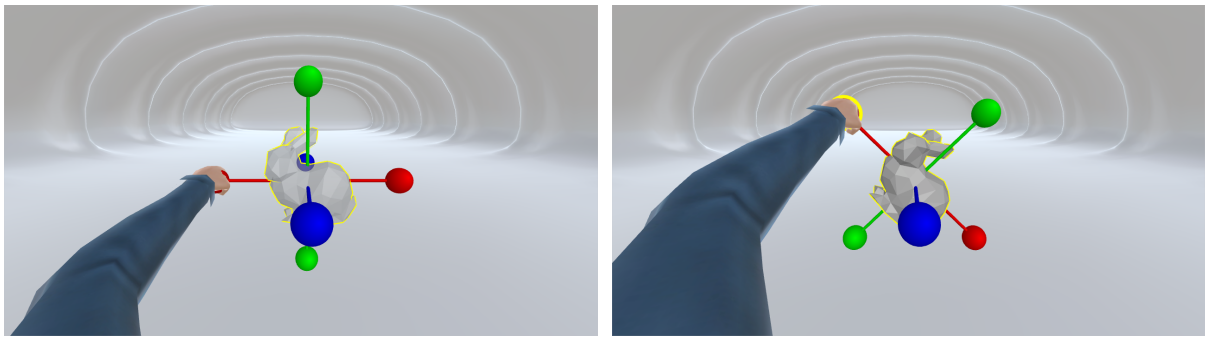


(a) Grab a single axis handle.



(b) Translate the object along the axis.

Figure 5.4: Widgets scaled translation.



(a) Grab a single axis handle. (b) Rotate the object around the axis.

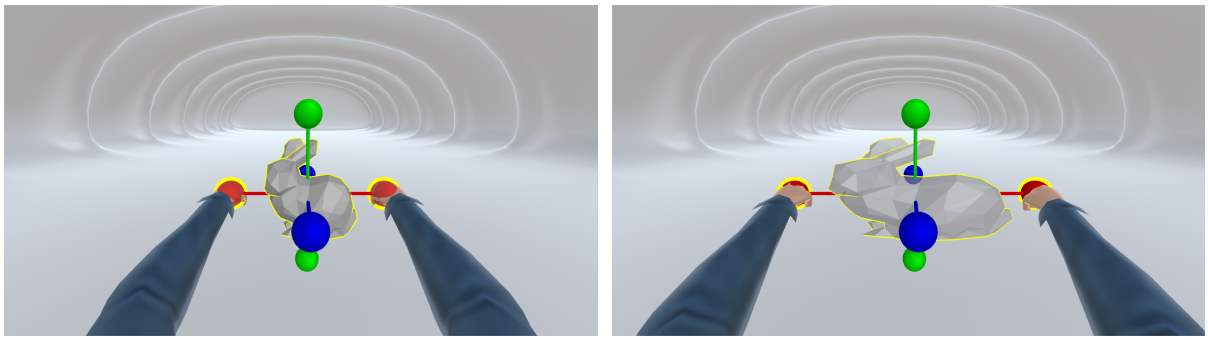
Figure 5.5: Widgets rotation.

5.3.2 Rotation

Similarly to translation, rotation shares the same behaviour regarding both decision regions and decision distance. We start by tracking users hand position from the previous and current frame. These positions are projected on the plane defined by the vector perpendicular to the rotation direction. By doing this we are able to calculate the angle every frame and users can use the distance to the plane to make small or big adjustments during rotation. As a result of this implementation, the closer the hand is to the plane the more direct is the transformation, while increasing its distance reduces the angle of rotation. This benefits precision as users can control the amount of rotation being applied to the object at a given time (Fig. 5.5).

5.3.3 Scale

WISDOM provides non-uniform and uniform scaling (1DOF and 3DOF) by making use of both hands. In order to perform uniform scaling users need to grab directly the object with one hand and pinch any place outside the object with the other hand at the same time. The initial distance after triggering this behaviour is taken as the starting point. Any increase or decrease in the distance between both hands is used to scale the object up or down respectively. Non uniform scaling (1DOF) relies on both hands as well, but users are required to grab two handles from the same axis. Moving both hands apart scales the



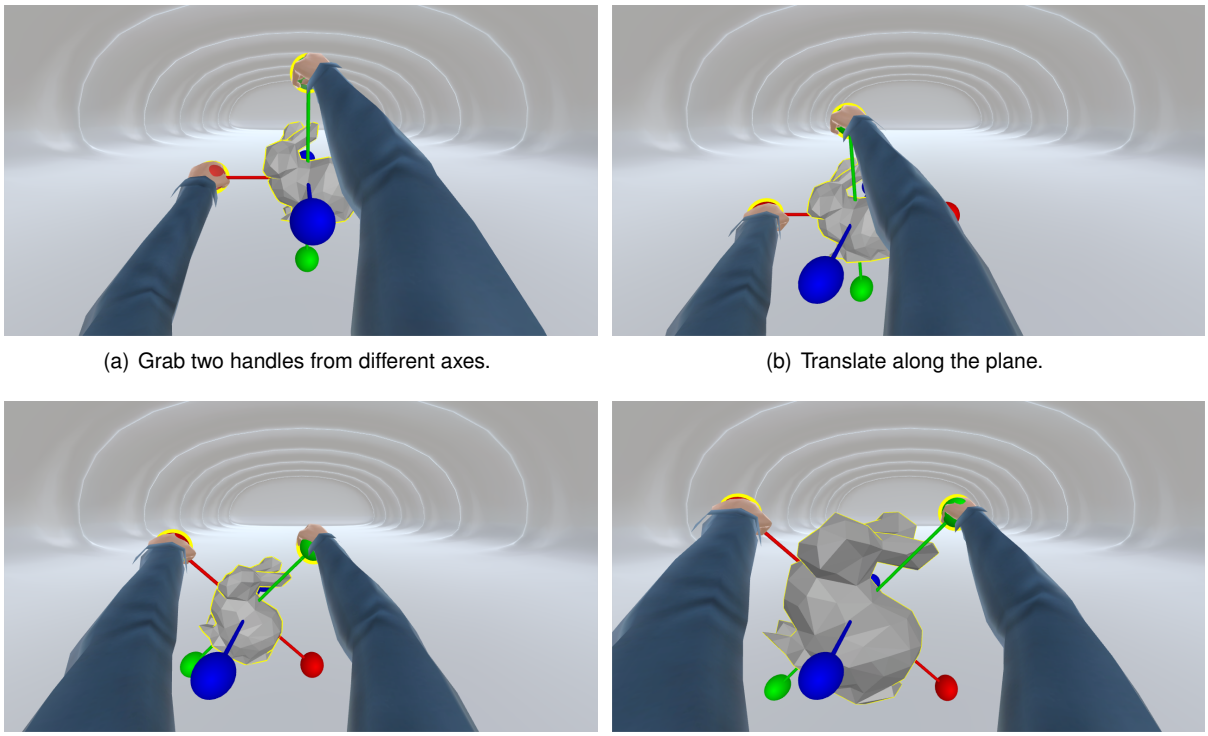
(a) Grab two handles from the same axis. (b) Scale along the axis.

Figure 5.6: Scale.

object up while moving them together scales it down along the axis defined by both handles (Fig. 5.6). We calculate the ratio between the current and previous hand positions which is later multiplied by the current object scale. Our prototype also provides non-uniform 2DOF scaling in the form of 2D TRS as described in the next subsection.

5.3.4 2D TRS

We implemented a 2D TRS [36] mode, which is enabled by grabbing two handles from different axes (Fig. 5.7(a)). We implemented this technique as it offers more freedom than 1DOF manipulation but maintains some constraints. While users can apply any transformation to the object at a given time, they are restricted to a certain plane. This plane can be seen as a table or a wall metaphorically and is represented by the two previously selected axes with the remaining one acting as a normal. For translation we calculate the difference between the current and last hands mid-point and move the object in the resulting distance and direction (Fig. 5.7(b)). For rotation we calculate the current and last distance vector between both hands in order to obtain the angle, while the rotation axis is defined by the plane normal (Fig. 5.7(c)). Regarding scaling, we use the distance between both hands similarly to object scaling, but with the possibility to scale along two axes as depicted in Figure 5.7(d).



(a) Grab two handles from different axes.

(b) Translate along the plane.

(c) Rotate around the plane normal.

(d) Scale along both axes.

Figure 5.7: 2D TRS.

5.4 Undo

Our proposal also offers an undo feature. A significant number of users complained about the inability of reverting any transformation applied to the object. We took this feedback into consideration and implemented a simple undo. This function is triggered by reaching the second pressure level while manipulating the object and allow users to revert any transformation being applied to it. If a release gesture is performed the decision is considered to be final and the transformation is stored to enable future corrections.

5.5 Summary

This chapter began with an overview of our proposed approach. WISDOM features different aspects of the techniques implemented in our assessment and allows users to switch between different manipulation modes depending on their needs. We follow this overview with a in depth explanation about each mode whether is direct manipulation, scaled transformations or DOF separation. Scale along 1DOF, 2DOF and 3DOF, as well as 2D TRS are also detailed and present two modes relying on both users' hands.

Chapter 6

Evaluation

During the development of our work, we had two different user evaluation phases. The first phase focused on the importance of DOF separation in mid-air techniques already described in Section 4.4 while the second user evaluation was conducted in order to validate our solution.

Initially users were asked to perform a set of tasks using three different manipulation techniques to assess the benefits of DOF separation in mid-air. We took the results from this evaluation and developed a technique that combines the most relevant aspects of each technique. Our solution was also subjected to user validation, where we followed a similar pattern to the first evaluation phase. We compared it against two of the previously techniques through a set of tasks in the same environment.

6.1 Methodology

The same structure was followed across all user sessions, each lasting approximately 40 minutes (Tab. 6.1). We began by introducing to the participants the experiment they were about to perform, followed by a brief description of the techniques being evaluated. The techniques were performed in alternated order, 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, showed in Figure 6.1. Following the practice period, participants were asked to perform a set of four tasks, described in the next section. After completing each technique's tasks, participants fulfilled a questionnaire regarding distinct aspects of the interaction. The experiment concluded with a profiling questionnaire.

6.2 Tasks

Participants were asked to complete a set of four tasks for each technique. All consisted in a docking task [24, 11, 35], where participants had to place a Tetris L-shaped piece in the same position and orientation of a transparent copy. Only the original piece could be grabbed and transformed in our virtual

Stage	Duration
Experiment Introduction	5 min
Technique Description	10 min
Training Period	5 min
Task Execution	15 min
Questionnaire Completion	5 min
Total	40 min

Table 6.1: Methodology estimated stage duration.

environment. Each task was limited to a maximum of three minutes in order to prevent excessively long sessions. 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 6.2(a)), the object to be manipulated begun with the correct orientation, but with a wrong position along all three coordinates. The second task (Figure 6.2(b)) implied only rotation around an arbitrary axis, as the object was already in the right position. In the third task (Figure 6.2(c)) users were required to rotate the object around the Y axis and translated along both XX and ZZ axes. Finally, in the last task (Figure 6.2(d)), participants had to apply full 6 DOF transformations to the object. Although some tasks required only one kind of transformation, none was restricted.

6.3 Apparatus and Participants

The experiment was performed in our laboratory with a controlled environment (Figure 4.9), using the setup detailed in Section 3.2. We counted with the participation of 20 people (3 female), between the ages of 18 and 40 years old, with the great majority (65%) between 18 and 25. Most had at least a BSc degree (80%), while the remainder are finishing it. More than half (60%) rarely or have never experienced a VR setting, and 45% never used any kind of gesture recognition systems, such as XBox Kinect, Wii Remote or Playstation Move. Only 25% of participants use 3D modelling systems at least once a month.

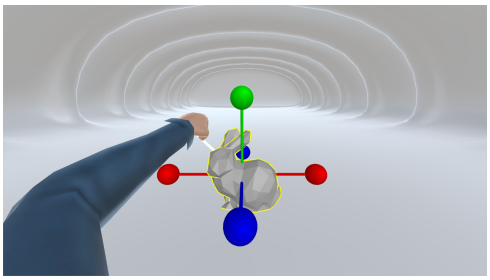
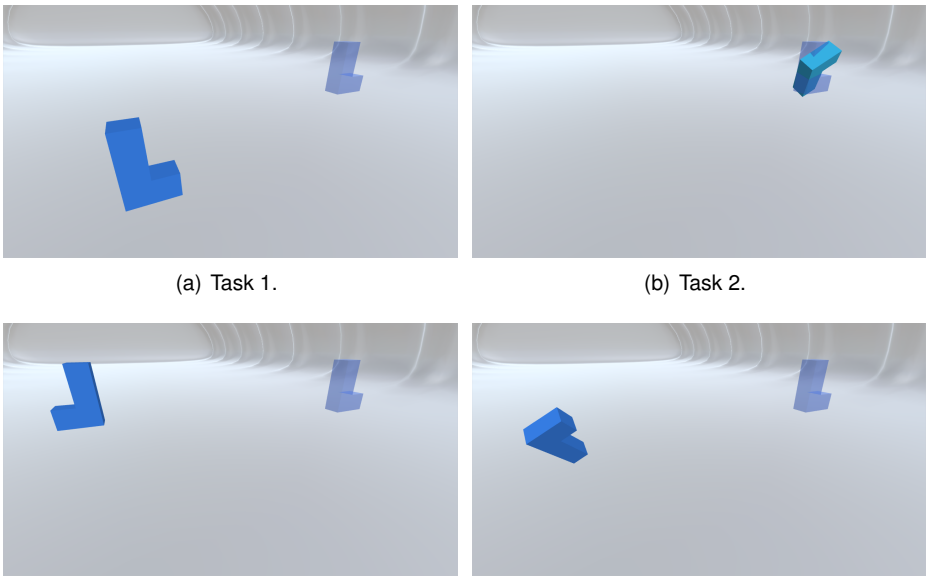


Figure 6.1: Interacting with an object in our virtual environment during the training period, with WISDOM technique.



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

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

Figure 6.2: Tasks performed by the participants.

6.4 Results and Discussion

We collected both objective data, through logging mechanisms, and subjective data, asking participants to fill out questionnaires, during our experiment. We used Shapiro-Wilk test to assess data normality. We then ran the repeated measures ANOVA test with a Greenhouse-Geisser correction to find significant differences in normal distributed data, and Friedman non-parametric test with Wilcoxon Signed-Ranks post-hoc test. In both cases, post-hoc tests used Bonferroni correction (corrected sig. = sig. $\times$ 3).

6.4.1 Objective Data

We measured time taken by participants to fulfil each task, as well as object placement error. Time taken for all tasks, in seconds, is depicted in the graph of Figure 6.3. Regarding errors, we registered both position error, in millimeters (Figure 6.4), and rotation error, in degrees (Figure 6.5).

For the translation only and rotation only tasks, we found statistically significant differences in completion time (Task 1: χ^2(2)=14.282, p=.001; Task 2: χ^2(2)=14.282, p=.001). For the first task, post-hoc test revealed 6DOF approach (avg=53.7s) to be faster than both Widgets (avg=108.4s, Z=-3.310, p=.003) and WISDOM (avg=117.8s, Z=-3.354, p=.003). In the second task, 6DOF (avg=49.1s) was also faster than Widgets (avg=107s, Z=-3.332, p=.003) and WISDOM (avg=113.5s, Z=-3.332, p=.003). The technique used also influenced rotation error in the first task (χ^2(2)=11.804, p=.003), with 6DOF (avg=5.8º) performing worse than Widgets (avg=3º, Z=-2.534, p=.033) and WISDOM (avg=3.1º, Z=-2.900, p=.012), and position error in the second task (χ^2(2)=7.964, p=.019) with WISDOM (avg=9.6mm) outperforming 6DOF (avg=16.3mm, Z=-2.605, p=.027).

Even though 6DOF was the faster 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

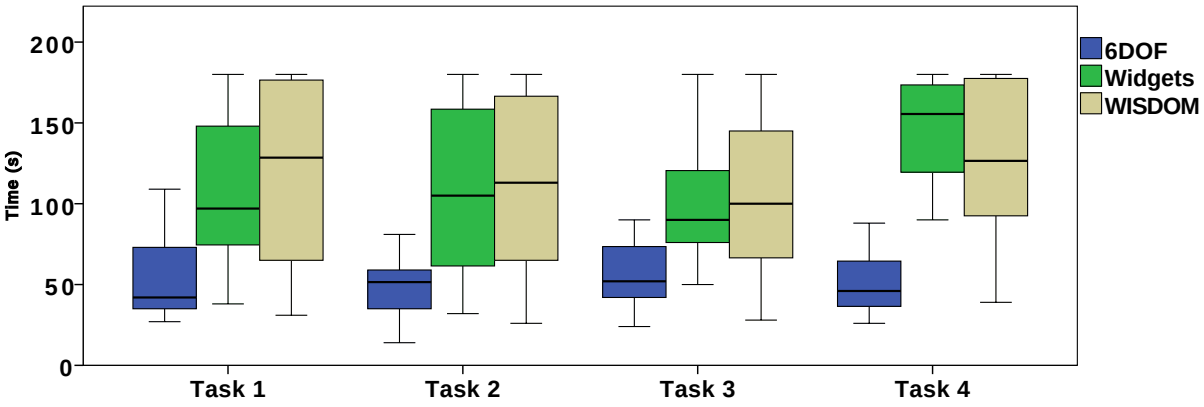


Figure 6.3: Time to complete the four tasks using the three techniques, in seconds. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

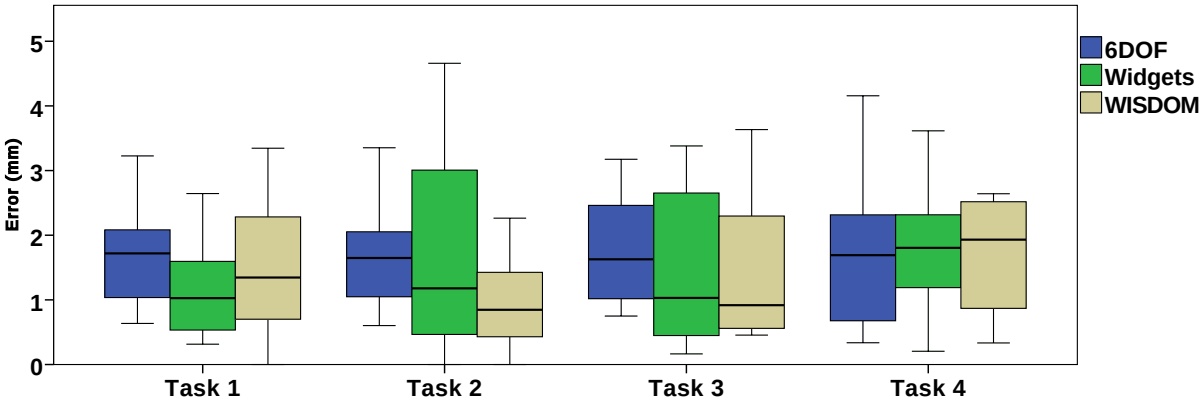


Figure 6.4: Position error attained in the four tasks using the three techniques, in millimeters. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

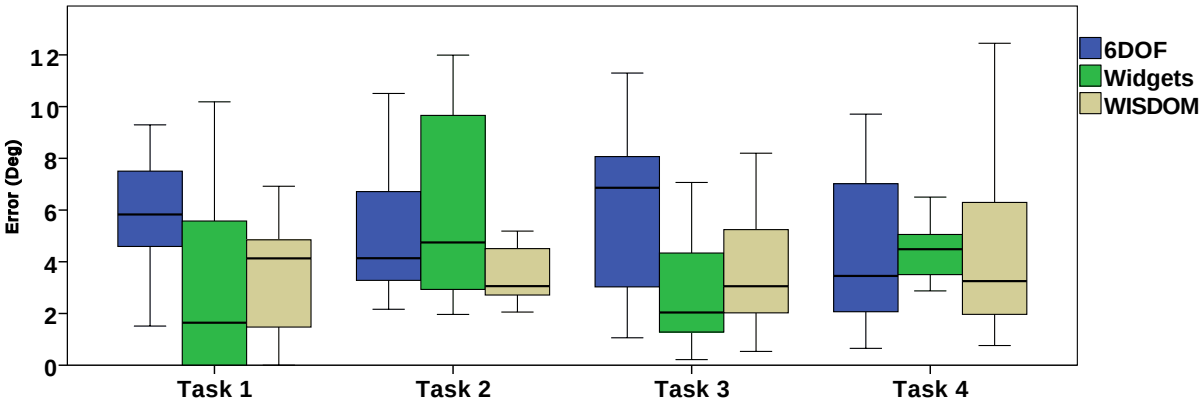


Figure 6.5: Rotation error attained in the four tasks using the three techniques, in degrees. The graphic presents the median, first and third inter quartile ranges (boxes) and 95% confidence interval (whiskers).

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 6DOF as opposed to the possibility of separating translation from rotation in WISDOM.

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 3: χ^2(2)=18, p<.0005; Task 4: χ^2(2)=24.108, p<.0005). 6DOF (avg=56.6s) outperformed Widgets (Task 3: avg=103.4s, Z=-3.623, p<.0005; Task 4: avg=146.2s, Z=-3.824, p<.0005) and WISDOM (Task 3: avg=106.8s, Z=-3.260, p=.003; Task 4: avg=124s, Z=-3.623, p<.0005) in both tasks. There were differences regarding error in object orientation in the third task (F(1.722,25.829)=14.436, p<.0005), where 6DOF (avg=6º) was outperformed by both Widgets (avg=2.7º, p=.001) and WISDOM (avg=3.6º, p=.007).

Final tasks increased complexity benefited 6DOF direct approach. 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 6DOF or Widgets. We observed that users used the time initially spared with 6DOF to make adjustments with Widgets after closing the distance and orientation to the target transformation. In the third task 6DOF presented less accurate rotations compared to Widgets and WISDOM. This might be related once again to the inability of separating transformations while performing tasks with 6DOF.

6.4.2 Subjective Data

We asked the participants how they felt about each technique using questionnaires. This included general easiness of use, translation/rotation difficulty and fun factor. Participants were given a Likert Scale from 1 to 5 to answer our questions, being 5 the favorable value. Answers are depicted in Table 6.2.

Analysing attained results, we identified no significant differences in any of the preferences. Ease of use (χ^2(2)=2.621, p=.270), fun factor (χ^2(2)=7.483, p=.024), translation difficulty (χ^2(2)=2.133, p=.344) and rotation difficulty (χ^2(2)=3.434, p<.0005) showed no statistical differences. None of the techniques achieved a median of five, meaning there's room for improvements across every criteria. Widgets produced a 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 6DOF and Widgets, WISDOM didn't present any significant improvements to users.

	6DOF	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 6.2: Participants preference for each technique, regarding different criteria (Median, Inter-quartile Range).

6.5 Discussion

At first we believed WISDOM would present lower completion times from providing both 6DOF and Widgets to users without compromising placement error. Our initial thought was to use 6DOF to place the object closely to the target and then switch to Widgets to increase object placement accuracy. In some cases we observed this didn't happen, as users took the time initially gained from a more direct manipulation to make adjustments in the end. As we did not impose any specific transformation order, users were able to make a decision based on their needs resulting in different approaches and distinct levels of performance. In more extreme situations, some users even took an exclusive approach to one of the two methods of manipulation, which might have caused some discrepancies in regards to time completion and object placement error.

As we mentioned earlier our technique also provided 2D TRS, which could be used at any time during tasks. However users forgot or didn't see the benefits of making use of it during object manipulation. It is important to notice that we included this feature in the explanation of our technique, but the large number of options and ways to interact with the object might have dissuaded users from making use of 2D TRS.

By comparing these results with the results from the DOF separation assessment we found significant differences regarding completion time and position error. Even though both translation-only and rotation-only single DOF tasks were removed, 6DOF as opposed to our initial evaluation was significantly faster than both Widgets and WISDOM in tasks requiring 3 DOF manipulation (Moderate tasks). Widgets also performed worse, in terms of position error, when it showed significant differences compared to direct manipulation on moderate tasks (3 DOF). Although we can't explain the reason for this behaviour, some of our proposed guidelines are called into question. Our third guideline, single DOF manipulation is very desirable for precise transformations, might be valid in certain instances namely simple tasks that only require 1 DOF. WISDOM which made use of scaled translation following our fourth guideline, didn't see any benefits in terms of position error, and while we believe this is a good approach for fine-grained adjustments the implementation found in PRISM might not be the best solution.

6.6 Summary

We conducted an evaluation to assess the benefits of DOF separation in mid-air, after it has been proved useful in other interaction paradigms by previous research. In this assessment we compared three mid-air object manipulation techniques, one follows a direct approach (6DOF), the second scales users' movement (PRISM) and the third is our implementation of mid-air virtual handles for DOF separation. Results showed that through Widgets, users were able to achieve lower error rates even though the completion time in more complex tasks was higher. We also drew some guidelines for future work in object manipulation in mid-air based on the results of the assessment.

Following the guidelines, we proposed a technique called WISDOM that combines different aspects of the techniques we used before. It provides the direct manipulation found in 6DOF and the scaled

movements we observed in PRISM, while maintaining DOF separation offered by our Widgets implementation. In order to validate our proposal we conducted another user testing. WISDOM improved position and rotation error in some instances compared to the other techniques, but didn't reduce time completion as opposed to our initial thoughts. It is also the first mid-air technique that provides 1DOF, 2DOF and 3DOF object scaling.

Chapter 7

Conclusions

As we are witnessing a huge interest in virtual reality (VR), mainly due to the recent technological advances that made head-mounted displays (HMD) affordable and widely available, mid-air interactions became a subject of interest in various research topics. Immersive virtual environments (IVE) that were made possible with such technologies are being used for several purposes, like engineering, architecture, game development and so forth, offering unique capabilities. To interact within those virtual environments (VE), the ability to manipulate virtual objects is a key feature.

Given its importance, object manipulation in virtual environments has been subject of research for long, covering different kinds of interaction paradigms. While direct approaches that mimic interactions in the physical world are the most natural, as the dimensional difference between input and output does not exist, it is still difficult to place a virtual object in the desired place with a high degree 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. For both mouse and touch interfaces, separation of degrees-of-freedom (DOF) led to better users' performance when compared to direct approaches, mainly due to the required mapping between the 2D input and 3D output.

In this work, we conducted an evaluation to assess the benefits of DOF separation in mid-air, after it has been proved useful in other interaction paradigms by previous research. We implemented three mid-air object manipulation techniques, one follows a direct approach (6DOF), the second scales users' movement (PRISM) and the third is our implementation of mid-air virtual handles for DOF separation. Regarding our first research question we concluded that single DOF separation through virtual widgets led to error reduction in 1 DOF tasks. Concerning the second question single DOF manipulations are as fast as direct approaches in 1 DOF and 3 DOF tasks. Drawn from our results we drew some guidelines for future work on object manipulation in mid-air.

Our prototype meshed together different devices and technologies. We used multiple body tracking sensors for user tracking, a custom device with an IMU for hand rotation tracking and an HMD to display our scene. We implemented a client in each device to handle data and send it to our prototype, running a server.

Following the guidelines we drew from the DOF separation assessment, we proposed a technique

called WISDOM that combines different aspects of the techniques we used to assess the importance of DOF separation in mid-air. It provides the direct manipulation found in 6DOF and the scaled movements we observed in PRISM, while maintaining DOF separation offered by our Widgets implementation. In order to validate our proposal we also conducted a user testing in this work. WISDOM improved position and rotation error in some instances compared to the other techniques, but didn't reduce time completion as opposed to our initial thoughts. It is also the first mid-air technique that provides 1 DOF, 2 DOF and 3 DOF object scaling.

7.1 Future Work

As a result of the work presented in this dissertation, we believe some aspects should be further studied as they might benefit future work. We describe the following:

- **Apply object transformations using mid-air hand gestures.**

  Our technique originated some improvements namely position and rotation error in certain tasks, but failed at reducing time completion. A distinct application from our guidelines might be an approach based on mid-air hand gestures to specify the desired transformations.

- **Define custom arbitrary transformation axis.**

  We believe it might be interesting to allow users to define custom arbitrary transformation as an addition to the Widgets approach. By doing so, object manipulation keeps the DOF separation benefits, but reduces substantially the number of operations needed. This will possibly allow for quick transformations in complex tasks.

- **Validate WISDOM regarding uniform and non-uniform scaling in mid-air.**

  Our technique provides 1DOF, 2DOF and 3DOF scaling but it wasn't enabled during user testing, since the techniques we compared it with didn't provide this capability. It is important to validate scale as it might be a compelling addition for object manipulation in mid-air.

Bibliography

[1] M. T. Schultheis and A. A. Rizzo. The application of virtual reality technology in rehabilitation. *Rehabilitation Psychology*, 46(3):296, 2001.

[2] L. Gregg and N. Tarrier. Virtual reality in mental health. *Social psychiatry and psychiatric epidemiology*, 42(5):343–354, 2007.

[3] 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*, pages 549–558. ACM, 2011.

[4] 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. ISBN 0-89791-513-5. doi: 10.1145/142750.142772. URL `http://doi.acm.org/10.1145/142750.142772`.

[5] 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.

[6] K. Shoemake. Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*, volume 92, pages 151–156, 1992.

[7] R. Schmidt, K. Singh, and R. Balakrishnan. Sketching and composing widgets for 3d manipulation. In *Computer Graphics Forum*, volume 27, pages 301–310. Wiley Online Library, 2008.

[8] 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*, pages 133–140. ACM, 2009.

[9] 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*, pages 69–78. ACM, 2009.

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

[11] 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*, pages 111–118. ACM, 2010.

[12] 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. ISBN 978-1-4503-0827-4. doi: 10.1145/2071423.2071447. URL `http://doi.acm.org/10.1145/2071423.2071447`.

[13] 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*, pages 3005–3008. ACM, 2011.

[14] 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.

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

[16] 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.

[17] 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*. IEEE, 2007.

[18] 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. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047273. URL `http://doi.acm.org/10.1145/2047196.2047273`.

[19] 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. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076377. URL `http://doi.acm.org/10.1145/2076354.2076377`.

[20] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff. ACM, 1997.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] W. A. König, J. Gerken, S. Dierdorf, and H. Reiterer. Adaptive pointing–design and evaluation of a precision enhancing technique for absolute pointing devices. In *Human-Computer Interaction–INTERACT 2009*, pages 658–671. Springer, 2009.

[27] 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.

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

[29] L. Gallo and A. Minutolo. Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique. *International Journal of Human-Computer Studies*, 70(4):287–300, 2012.

[30] 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*, pages 139–148. ACM, 2009.

[31] T. Ha and W. Woo. An empirical evaluation of virtual hand techniques for 3d object manipulation in a tangible augmented reality environment. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 91–98. IEEE, 2010.

[32] 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.

[33] 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.

[34] 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 SIGCHI Conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

[35] D. Mendes, F. Fonseca, B. Araujo, A. Ferreira, and J. Jorge. Mid-air interactions above stereoscopic interactive tables. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 3–10. IEEE, 2014.

[36] M. S. Hancock, F. D. Vernier, D. Wigdor, S. Carpendale, and C. Shen. Rotation and translation mechanisms for tabletop interaction. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06)*, pages 8–pp. IEEE, 2006.