A Survey on 3D Virtual Object Manipulation: from the Desktop to Immersive Virtual Environments

D. Mendes[1], F. M. Caputo[2], A. Giachetti[2], A. Ferreira[1] and J. Jorge[1]

[1]INESC-ID Lisboa, Instituto Superior Técnico, Universidade de Lisboa
[2]Università degli Studi di Verona

Abstract

Interactions within virtual environments often require manipulating 3D virtual objects. To this end, researchers have endeavoured to find efficient solutions using either traditional input devices or focusing on different input modalities, such as touch and mid-air gestures. Different virtual environments and diverse input modalities present specific issues to control object position, orientation and scaling: traditional mouse input, for example, presents non-trivial challenges because of the need to map between 2D input and 3D actions. While interactive surfaces enable more natural approaches, they still require smart mappings. Mid-air gestures can be exploited to offer natural manipulations mimicking interactions with physical objects. However, these approaches often lack precision and control. All these issues and many others have been addressed in a large body of work.
In this article, we survey the state-of-the-art in 3D object manipulation, ranging from traditional desktop approaches to touch and mid-air interfaces, to interact in diverse virtual environments. We propose a new taxonomy to better classify manipulation properties. Using our taxonomy, we discuss the techniques presented in the surveyed literature, highlighting trends, guidelines and open challenges, that can be useful both to future research and to developers of 3D user interfaces.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles

1. Introduction

Since the early days of virtual environments (VEs), the search for effective methods for translating, rotating and resizing virtual objects has been a major research target. Considering three-dimensional VEs, these types of manipulations are not trivial, mainly due to the required mapping between traditional input devices (2D) and the virtual environment (3D). Most of the common solutions resort to techniques that somehow relate the actions performed in the two-dimensional space of the input device (e.g. mouse cursor or touch) to three-dimensional transformations.

Aiming to offer more natural interfaces, touch-enabled surfaces introduced the possibility of directly interacting with virtual content. Although having a 2D input similar to mouse-based interfaces, users are able to touch the virtual objects that they want to manipulate. Additionally, by allowing simultaneous touches, interfaces can have a higher input bandwidth, leading to new manipulation techniques.

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

Recent technological advances have led to an increased interest in immersive virtual reality settings. Affordable hardware for immersive visualization of VEs, such as the Oculus Rift, HTC Vive, Samsung Gear VR and Google Cardboard head-mounted displays (HMDs), ease the perception of three-dimensional content. Such hardware hinders the use of traditional input devices for interaction, but existing user tracking solutions make it possible to know where users' heads, limbs and hands are in space. Over the past years, novel non-intrusive and affordable spatial tracking solutions have been proposed. Such solutions allow for more direct and natural interactions, mimicking the interactions with physical objects.

Although mid-air interactions show promising results, the accuracy of human spatial interactions is limited. In fact, the limited dexterity of mid-air hand gestures, which is aggravated by the lack of precision from tracking systems and the low definition of current HMDs, reduces the precision of manipulations. Although an accurate object location is not required in some applications (e.g. visualization), precision is of extreme importance when creating or assembling engineering models or architectural mock-ups, for instance.

To approach the large diversity of manipulation approaches and technologies recently proposed for VEs, we present in this paper a survey of the related literature, reporting the principal aspects of the methods and classifying them according to selected criteria, to provide researchers with a useful tool to better understand the pros and cons and the potential of the different approaches. Although previous publications do cover 3D interactions, we present an up-to-date report of manipulation techniques. Bowman et al. [BKLJP04] compiled a comprehensive set of 3D interaction techniques and devices until the early 2000s. The revised version [LJKM*17], as well as [LaV17], discuss development issues and techniques proposed since the first edition, but because many topics are covered, the treatment of the subject matter is not as deep as our survey. Other recent surveys regarding interactions with VEs are also very broad [JH15], covering subjects such as navigation, selection, manipulation and evaluation techniques, or do not focus on the transformation part of object manipulation [AA13]. We will focus on this latter subject, presenting a thorough coverage of the literature.

The remainder of this paper is organized as follows: Section 2 presents a set of key concepts for object manipulation in virtual environments, introducing a new taxonomy for 3D object manipulation that is useful for organizing and discussing the reviewed literature. Starting in Section 3, we present the most relevant research works regarding 3D virtual object manipulation. We will first address traditional desktop interactions with screen-constrained visualization and mouse-based 2D input. Then, we cover touch-based manipulation with both screen-constrained visualization settings and stereoscopic tabletops in Section 4. Finally, in Section 5, we report on manipulations based on mid-air input. Section 6 presents a discussion, where we identify trends, guidelines and open issues. Throughout the paper, we mention and cite many manipulation techniques, some of which also feature demonstration videos to illustrate their functionality. To supplement this survey, we provide an updated collection of these videos available online at `http://web.ist.utl.pt/ist153804/survey3dom`, that we believe will help readers better understand the proposed interaction mechanisms in detail.

2. Key Concepts for Object Manipulations in Virtual Environments

VEs have been around for some time, and they are used for a myriad of purposes. Ranging from bioengineering and geology [VDLS02], oil and gas [GMM*14], automotive engineering [MF11b], manufacturing [MSH04], architectural mockup [ACJH13] and CAD [HZS*13] to creative painting [KFM*01], animation movies [MYC15] and entertainment with building blocks [MLF11], VEs are something that we now take for granted.

2.1. Overview of Virtual Environments' Inputs and Outputs

User immersion in VEs can be enhanced by combining stereoscopic visualization and head tracking. By knowing the user's head position, it is possible to generate a visualization frustum to each eye to create the illusion of virtual objects being part of the physical world. This illusion is even stronger when users are allowed to freely move their heads and see different sides of a virtual object in their own perspective, without the need to manipulate cameras or widgets.

Although HMDs and CAVEs (cave automatic virtual environments), which allow a fully immersive viewing experience, have existed for a while, interest in these technologies has increased considerably over the past few years. One of the main issues with older HMDs was the nausea that they caused, commonly referred to as virtual reality sickness or cybersickness.

However, the new generation of low-cost HMD devices that have recently appeared have demonstrated that this issue can be effectively solved by using low-latency inertial devices and smart rendering solutions such as the time-warping technique [DBB15].

Other recent technological advances have also made it easier to develop immersive visualization scenarios. Not long ago, user tracking required expensive and invasive systems. Currently, user tracking is possible using affordable and non-intrusive methods based on depth cameras, IR cameras and markers on headsets and low-latency inertial sensors. These tracking solutions can be used not only to find the user point of view to render the virtual scene but also to track user limbs and hands, unveiling new interaction possibilities. Additionally, this combination of stereoscopic displays and user tracking allows users to naturally manipulate three-dimensional entities as if they were co-located with their hands and body, extending traditional two-dimensional interactions in very natural ways.

A VE that can be explored through immersive displays is often called an immersive virtual environment (IVE) or a virtual reality (VR). Although a fully immersive environment should explore other human senses in addition to vision, as studied by Azevedo et al. [AJC14], the IVE classification is often used when using only an immersive display. According to Bowman et al. [BKLJP01], these types of displays can be divided into two categories: fully immersive displays, such as HMDs, which completely occlude the real world, and semi-immersive displays, such as stereo tabletops, which allow users to see both the physical and virtual worlds. The benefits of higher levels of immersion in virtual reality setups have already been presented [BM*07].

To describe the most relevant aspects of the VEs presented in the surveyed literature, we classify their properties following the organization proposed by Grossman and Wigdor in their analysis of tabletop interactions [GW07], adapted to generic environments.

Starting with the display properties, we distinguish conventional 2D displays from those providing stereoscopic depth cues regarding the space where imagery appears to be. We also differentiate this space from the actual space where the rendered images are presented. This is constrained to 2D for most of the current interaction setups because truly volumetric displays that illuminate voxels in mid-air are not common. When 3D perceived space is generated

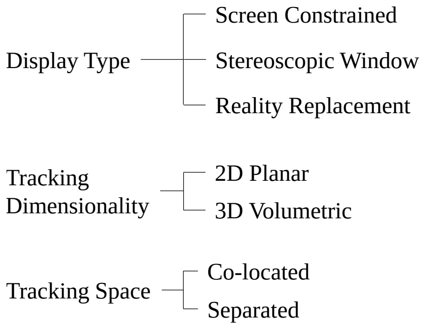


Figure 1: Overview of virtual environments' input and output properties.

on 2D screens with stereo and motion parallax cues, issues such as hand occlusions may arise. To overcome this issue, there are heads-up surfaces, such as HMDs or see-through screens placed between the user's eyes and hands.

Another important characteristic of the rendering setup is the viewpoint correlation, which concerns the relation between the user's point of view and the viewpoint of the virtual scene. In systems where the user moves around the display and the viewpoint remains constant, there is no relation. For systems that change the viewpoint of the rendering according to the user's head position, we say that there is a high or total correlation. High refers to setups composed of a screen, either vertical or horizontal, that when the user moves his head behind the screen, he will see the back of the screen rather than the VE from a different perspective. When using a HMD to create a virtual reality experience, total correlation between the user's viewpoint and the displayed imagery can be achieved.

Since there are several aspects that generally go hand-in-hand, according to different setups, we summarize some properties of virtual environments' inputs and outputs in Figure 1. Focusing on display type, VE visualization can be screen constrained, made through a stereoscopic window or be perceived as a reality replacement. Screen-constrained visualizations, such as those of traditional desktop displays, are based on rendering on 2D screens with no stereo depth cues and have no viewpoint correlation. Stereoscopic window, although also constrained to a 2D screen, offers a view of the VE with stereoscopic depth cues and high viewpoint correlation. With this visualization, virtual objects can appear to be within the screen (positive parallax), generally referred to using a fish tank metaphor; at the screen plane (zero parallax); or between the user and the screen (negative parallax). Using heads-up surfaces, fully immersive displays have total viewpoint correlation and employ stereo depth cues, replacing users' reality with the virtual one.

In these VEs, user interaction is often leveraged by tracking handheld devices or human body parts in 2D (e.g. mouse, touchscreens) or in 3D (through inertial or vision-based trackers). The tracking system may also allow the co-registration of the visualized and input spaces, allowing direct interactions with the virtual content. Despite acknowledging that there are several multi-modal interaction techniques that resort, for instance, to speech and/or gaze in addition to the aforementioned gestural input (e.g. [Bol80, SBAG16]), we will shorten the spectrum of this survey by focusing mostly on hand-based techniques. These techniques can be either hands-free through multi-touch and mid-air input or through handheld devices, such as mouse or spatially tracked controllers.

2.2. Manipulation: Transformations and Degrees of Freedom

We are interested in a particular type of interaction in VEs: the manipulation of existing virtual objects in the scene. Manipulation is the task of changing the characteristics of a selected object [BH99]. For instance, it can be considered as an application of spatial transformations; change of visual properties, such as colour or texture; or even free-form deformation. However, manipulations most commonly refer to spatial transformations [BH99, BKLJP04].

Several different types of spatial transformations exist: translation, rotation, scaling, shearing, and reflection, among others. Although there are research works that cover all these transformations in VEs, the most common transformations are translation and rotation, which are required for positioning tasks. These transformations are also the ones that have the greatest resemblance to everyday physical interactions. Nonetheless, since the seminal works [NOJ87, ZFS97], scaling has been grouped with these two basic operations. This trio of transformations, identified as the basic manipulation tasks [BKLJP04] along with selection, has been kept together in a plethora of other research works; several are described in this document, whereas the remaining transformations are not considered. Moreover, these three transformations generally appear together in commercial 3D software, such as Blender and Unity3D.

Positioning manipulations can be performed in diverse ways, either on a single object in isolation disregarding its surroundings, or by aligning and snapping to other objects in the scene [Bie90, SSB08], or even by grabbing multiple objects and aligning them, either packing or evenly distributing, or simultaneously moving them [SGH*12]. Ultimately, however, "any 3D manipulation can be constructed by translations and rotations around the object origin" [SSB08]. Therefore, we will focus on the basic canonical manipulation tasks, namely translation, rotation and scale [BKLJP04], of single virtual objects.

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

A manipulation requires a preceding selection and a release afterwards [BH99]. Depending on the characteristics of the virtual environment and its objects, different selection strategies can be followed [AA13]. For example, specific selection tools can ease

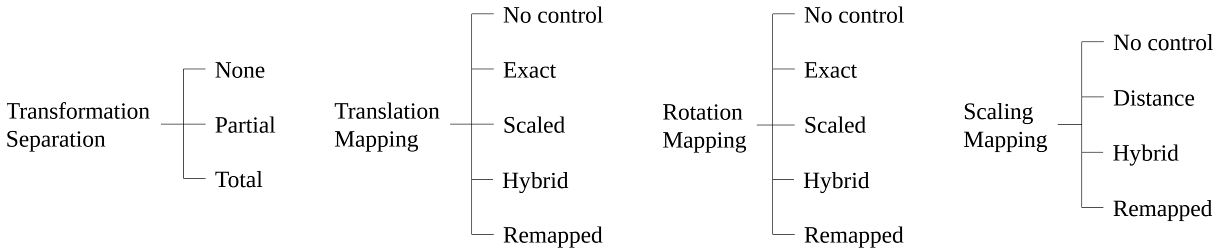


Figure 2: Taxonomy for classifying different approaches for object spatial transformations in virtual environments.

selection in clustered environments but will possibly require some disambiguation mechanism, and using different control-display ratios can either help performing out-of-reach selections or increase precision for small objects. Several release strategies also exist [BH99] and play a relevant role because they can lead to unwanted outcomes if poorly chosen. For instance, virtual objects may still be moved before the system detects a user's open hand or lifted fingers. Although these actions can affect performance in manipulation tasks, we will not cover selection and release strategies to restrict the length and scope of this document.

2.3. Mappings and Remappings of Transformations

Transformations are enabled by *mapping* a user's input onto actions performed on the manipulated object. This can be performed either through physics simulation or pre-programming specific interaction behaviour using, for example, gesture recognition. For the first, user input can be mapped to contact forces due to friction and collisions from virtual proxies within the physics simulation, enabling emergent hand-based gestures [WIH*08, HKI*12]. These can include, for instance, sweeping, scooping, lifting, and throwing virtual objects [HKI*12]. However, "some aspects of traditional interactions do not naturally lend themselves to a physics implementation" [WIH*08]. For example, dynamically scaling an object cannot be implemented through a rigid-body simulation. Consequently, we will focus on pre-programmed interactions, where input DOFs are explicitly mapped onto manipulated object transformation DOFs. Input DOFs can be those derived by tracking position and orientation in 3D, but they can also be measurements of user actions obtained through other input channels (buttons, trackballs, and isotonic and elastic sensors [Fro05]).

To classify the different mappings used in virtual object manipulation in VEs, we developed a taxonomy, presented in Figure 2, based on that proposed by Bowman and Hodges [BH99]. Whereas Bowman and Hodges covered all steps involved in a manipulation task (selection, transformation and release), we focus solely on transformations and go further in this component. Additionally, transformations can be applied simultaneously, having *no* separability, as it occurs with physical manipulations (translation and rotation) and common multi-touch interactions where users can move, rotate and scale objects with a single gesture. Transformations can also be applied separately, as is common in 3D modelling/editing software, requiring users to apply a single transformation at a time. However, some manipulation techniques group different transformations or only some DOFs from a transformation type, while separating the others. For these, we refer to them as having *partial* transformation separability. We only consider a technique to have *total* transformation separability when it enables users to perform every supported transformation in isolation from the others, e.g. it is possible to move an object to a new position along all axes without performing a single modification to its orientation or scale.

To map users' input onto transformations, several approaches can be followed. An *exact* manipulation maps the spatial transform of a device or a hand tracked directly onto the virtual object transform. In other words, it offers a 1:1 control, even if the tracked input and the virtual object have a fixed offset. If the tracked hand/device is co-located with the virtual one, then the effect is a simulation of a real-world manipulation. The selected translational or orientation DOF of the tracked input can also be mapped directly onto the virtual world's ones or with a linear or nonlinear *scaled* transform to increase accuracy or obtain increased ranges of transform parameters through N:1 or 1:N controls, respectively.

To overcome the limitations due to physical constraints, to allow 3D manipulation with 2D tracking, or to limit manipulated DOFs when having higher input DOFs, many techniques rely on indirect mappings. These mappings map tracked DOFs onto different manipulation DOFs (e.g. a slider controlling rotation, virtual widgets applying restrictions or specific gestures outside the object to trigger additional transformations) or use different input channels to control object transform DOFs (e.g. mouse, keyboards, joysticks, microphones, and so forth). This *remapping* procedure might involve "learning a sensorimotor mapping that produces different results in a virtual world than one would expect from the real world" [LaV17], and it is probably the most critical design issue in the development of a manipulation technique because it is difficult to find an optimal solution for different contexts. Mapping should allow the user to exploit existing or easy to learn motor programs [LaV17], making the interaction effective, easy to learn and easy to use. This result is often searched through the use of metaphors.

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

Direct manipulation with an exact mapping is not always the best solution, particularly when pursuing maximum accuracy or when we want to allow large translation, rotation and scaling. It is generally desired in immersive VEs, but it is perfectly acceptable, even in such environments, to remap different tracked motions or actions on buttons or joysticks onto object transforms. This is a typical solution in modern VR games, for example. We will discuss design choices in Section 6.

For scaling, since an exact mapping does not exist because this transformation is not physically possible, the most common is a *distance* mapping. This mapping resorts to the variation in the distance between two input points, using the metaphor of "stretching a piece of rubber" [ZFS97]. This mapping was suggested long ago [NOJ87] and made popular to the common public with the advent of touch-enabled mobile devices. However, different approaches for performing scaling transformation exist, which remap the input differently.

In the following sections, we introduce the vast amount of manipulation methods proposed in the literature that exploit 2D tracking, multi-DOF devices and 3D tracking techniques. In the beginning of each section, we summarize the reviewed techniques in a table, classifying them according to the taxonomy presented in Fig 2. In addition to the taxonomy concepts, we also characterize each technique regarding environment properties presented in Section 2. For techniques that were further developed, leading to a new and improved technique, we only consider their latest stage (e.g. touch-based Z-Technique [MCG10a] and DS3 [MCG10b], and mid-air 3-Point++ [ND13] and 7-Handle [NDP14]). We use abbreviations for display properties (SC: screen constrained, SW: stereo window, and RR: reality replacement). We also identify whether a technique separates transformations and, for each transformation type, its mapping, the number of required contact points (CP), total transformation DOFs supported (TD) and the minimum explicitly simultaneously controlled DOFs (MD). Additionally, regarding transformation separation, we indicate which transformations are grouped or set apart, e.g. {T,R,S} means that translation, rotation and scaling operations are applied simultaneously, whereas {T},{R},{S} indicates that it is possible to fully control all supported DOFs of each transformation separately from the other transformation. Although we do not go into further detail on which DOFs of each transformation are controlled together, we occasionally separate DOFs from a transformation to clarify how the separation is performed. For instance, {Txy},{T,Rz} means that translations on both x and y axes are applied simultaneously, but to also translate in the z axis, users must enable rotations around the same axis.

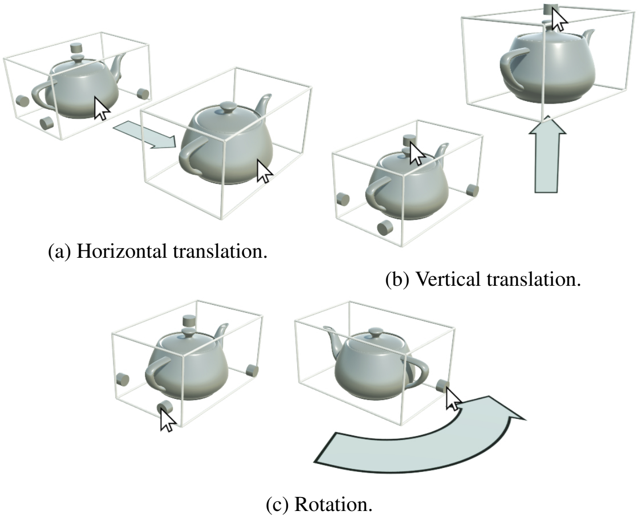
3. 3D Interactions based on Mouse and Keyboard

Many computer applications, such as architectural modelling, virtual model exploration, engineering component design and assembly, require virtual three-dimensional object manipulations, among others. To work with VEs for this purpose, several interaction techniques for traditional desktop setups have been explored, resorting to mouse and keyboard devices. Table 1 summarizes the techniques we will present in this section.

To overcome the mapping of 2D mouse input to 3D, Nielsen and Olsen [NOJ87] created the triad cursor. Mapping is performed by comparing its screen-space movements with the projected image of its three perpendicular axes. By also taking advantage of the projections of the object's features, it allows separate translation, rotation and scaling transformations according to the object's edge or a plane defined by a face of the object. Zeleznik et al. [ZFS97] used two cursors, one controlled by each hand, to simultaneously perform the three different transformations restricted to a pre-specified plane in 3D.

Alternatively, Stephanie Houde developed an approach based on a handle box [Hou92]. This approach consisted of a bounding box surrounding the object, and it had a lifting handle attached to it to move the object up and down and four rotation handles to rotate the object about its central axis, as illustrated in Figure 3. No handle was provided for sliding in the object's resting plane, on the assumption that the most direct way to slide an object would be to click and drag on the object inside the box itself. Conner et al. [CSH*92] also resorted to virtual handles to develop 3D widgets for performing transformations on virtual objects. They allow full 9-DOF control (translation, rotation and scaling) and even other deformations, such as twisting. The handles have a small sphere at their ends, and they are used to constrain geometric transformations to a single plane or axis (Figure 4). Dragging one of the spheres can translate, rotate or scale the object depending on which mouse button is pressed. For rotations, the direction of the user's initial gesture determines which of the two axes perpendicular to the handle is used as the rotation's axis.

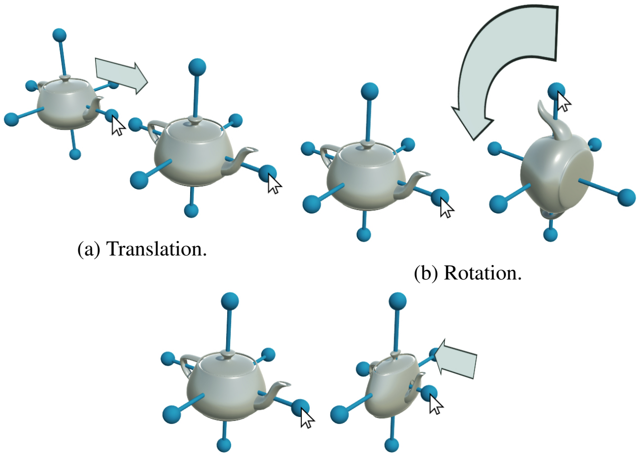
Focusing only on rotations, Ken Shoemake proposed Arcball [Sho92], an input technique that uses a mouse to adjust the spatial orientation of an object. To change the object's orientation, the user draws an arc on a screen projection of a sphere. For axis-constrained rotations, Arcball includes the view coordinate axes, the selected object's model space coordinate axes, world space coordinate axes, normals and edges of surfaces, and joint axes of articulated models (such as robot arms). Mouse, menu, or keyboard combinations can be used to select among axis sets. As an example, for body coordinate axes, three mutually perpendicular arcs


(a) Horizontal translation.

(b) Vertical translation.

(c) Rotation.

Figure 3: Manipulation using the handle box approach [Hou92].



(a) Translation.

(b) Rotation.

(c) Scaling.

Figure 4: Virtual handles for object manipulation [CSH*92].

would be drawn, which are tilted with the object. When the mouse is clicked down to initiate a rotation, the constraint axis selected will be that of the nearest arc.

More than 20 years have passed since these techniques were proposed, and they are still currently being used in several solutions, even commercial ones. Indeed, some applications that require object manipulation, such as Unity3D or SketchUp, resort to widgets both for mapping between input devices and corresponding 3D transformations and for restricting DOF manipulation. For interactively translating and scaling virtual objects, Unity3D, a commonly used game engine, allows users to do so through virtual handles, as depicted in Figure 5, similar to Conner et al. [CSH*92]. For rotations, it uses a direct implementation of Arcball [Sho92]. SketchUp, a 3D modelling application, resorts to a handle box for object scaling, as also shown in Figure 5. It provides quick and accurate modelling, aided by dynamic snapping, input of exact values for distances, angles and radius. All these solutions allow users to perform a transformation in a single axis at a time.

Other commercial applications, namely, those for 3D modelling, often present a different option. Rather than using widgets to restrict DOF manipulation, they allow the 3D VE to be presented through three orthogonal views. Examples of this are 3D Studio Max or Blender (Figure 6). In this way, each view allows simple 2D manipulations along different axes, overcoming mapping issues. However, they require users to have greater spatial perception, rendering them suitable only for expert users. AutoCAD, which is more focused on architectural and engineering projects, also features these orthogonal viewports and allows for extremely precise manipulation of the elements within the VE.

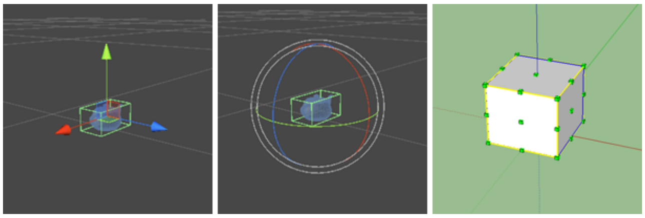


Figure 5: Widgets used in current commercial applications: virtual handles (left) and Arcball (middle) in Unity3D; handle box (right) in SketchUp.

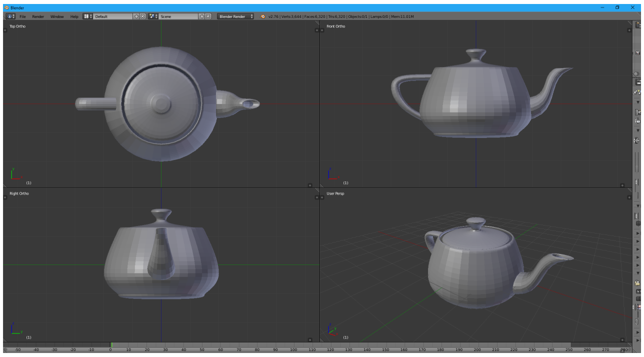


Figure 6: Orthogonal viewports in Blender.

3.1. Multi-DOF controllers (non-tracked in 3D)

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

SpaceMouse and other products by 3Dconnexion [3dc17] are probably the most known examples and are also a commercial success, as they are used in CAD applications, visualization and are compatible with many related desktop application packages. These devices allow users to manipulate a pressure-sensitive handle to manipulate 3D models within an application. They allow to pan, zoom and rotate 3D objects simultaneously without external actions.

GlobeFish and GlobeMouse [FHSH06] are other experimental multi-DOF mapping devices. "The GlobeFish consists of a custom three degrees of freedom trackball which is elastically connected to a frame. The trackball is accessible from the top and bottom and can be moved slightly in all spatial directions by applying force. The GlobeMouse device works in a similar way. Here the trackball is placed on top of a movable base, which requires to change the grip on the device to switch between rotating the trackball and moving the base."

CAT [HGR03] is another experimental 6-DOF freestanding device. It consists of a round tabletop that can be rotated about its three axes and features a movable ring around it connected to dynamometers that able to check pressure applied in all three directions. Roly-Poly Mouse [PSR*15] attempts to combine the advantages of devices such as SpaceMouse for 3D pointing and manipulation tasks with the functions of a standard mouse when a 2D pointing task has to be performed.

4. 3D Manipulation on Interactive Surfaces

Beyond the traditional WIMP-based approaches (windows, icons, menus and pointing devices), several multi-touch solutions to manipulate 3D objects have been proposed and evaluated over the past few years. In fact, touch-enabled displays have long been available, but their increased interest occurred following Jeff Han's work [Han05] and his acclaimed TED talk. With these interactive surfaces, new interaction possibilities emerged, allowing researchers to explore more natural user interfaces (NUIs) [WW11]. Efforts have been directed towards attempting to create more direct interactions with virtual content, closer to the ones with physical objects, which can successfully surpass mouse-based interactions [KAD09]. Touch-enabled surfaces are now present in our everyday life through smartphones and tablets. Interactive tabletops are also becoming increasingly more popular. These types of surfaces have been used for a variety of purposes, including interacting with 3D virtual content. The manipulation techniques we will review in this section are summarized in Table 2.

4.1. Direct Touch Manipulations

Since it has been shown that rotation and translation have a parallel and interdependent structure in the human mind [WMSB98], studies initially proposed techniques for controlling several DOFs simultaneously. Hancock et al. [HCC07] developed techniques to control 6 DOFs using one, two and three touches. The authors started by extending the Rotate'N Translate (RNT) algorithm [KCST05] to the third dimension. When touching an object, that object will follow the finger, rotating along all three axes and translating in two dimensions. Using two touches, the original two-dimensional RNT is used with the first touch, while the second touch rotates the object in the remaining axes. The distance between the two touches changes the depth of the object. The three-touch approach uses the first contact point for translations in a two-dimensional plane, the second to yaw and manipulate depth, and the third to pitch and roll. After evaluating this technique, the authors concluded that a higher number of touches provides both better performance and higher user satisfaction. These results suggest that a close mapping of input and output DOFs is desirable. The authors also defined a set of requirements for multi-touch interfaces, such as creating a visual and physical link with objects and providing suitable 3D visual feedback. Later, they improved the proposed techniques with Sticky Fingers and Opposable Thumb [HtCC09]. This solution is very similar to the three-touch technique, but in this solution, the third touch is used to rotate the object around the axis defined by the first two touches (Figure 7).

Considering the *de facto* standard for 2D manipulations, the Translate-Rotate-Scale (TRS) or two-point rotation and translation with scaling [HVW*06], Reisman et al. [RDH09] proposed a screen-space formulation that uses several points of contact in a multi-touch device to manipulate 3D objects in 6 DOFs. Similar to previous works, rather than supporting scaling transformations, the distance between contact points is mapped to depth manipulation according to the view vector. The rationale is that the object appears larger when it is closer to the camera and smaller otherwise. Their solution keeps the contact points fixed throughout the interaction, using a constraint solver to move and rotate objects simultaneously. This solution is similar to Opposable Thumb, but if the movement of the third finger is not perpendicular to the defined axis, then that axis is no longer used and the object will rotate to follow the finger. The main issue of providing an integrated solution to manipulate different transformations simultaneously is that unwanted operations arise frequently. To remedy this issue, the separation of DOF manipulation has been suggested [NBBW09] and followed in different research works.

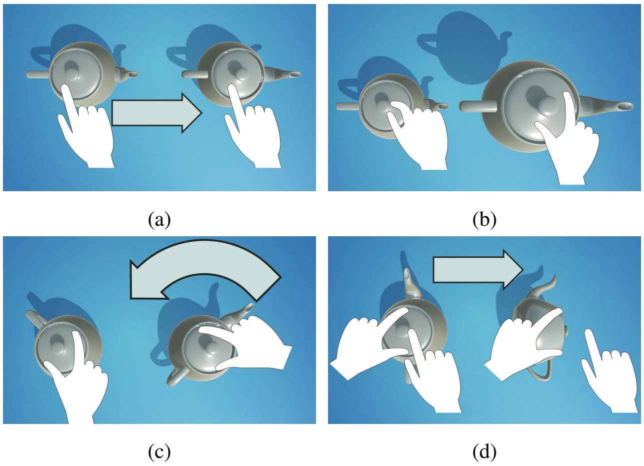

(a)
(b)
(c)
(d)

Figure 7: Sticky Fingers (a, b, c) and Opposable Thumb (d) [HtCC09].

Martinet et al. [MCG10a] proposed two techniques to translate 3D objects. The first extends the viewport concept found in many CAD applications (four viewports, each displaying a different view of the model). Touching and dragging the object within one of the viewports translates the object in a plane parallel to that view. Manipulating the object with a second touch in a different viewport modifies depth relative to the first touch. For the second method, denoted as the Z-technique, only one view of the scene is employed. In this technique, the first touch moves the object in the plane parallel to the view, while the backward-forward motion of a second touch is remapped to control the depth relative to the camera position, as shown in Figure 8. The authors' preliminary evaluation suggests that users prefer the Z-technique.

Improving upon the Z-Technique, Martinet et al. introduced DS3 [MCG10b], a 3D manipulation technique based on DOF separation. Similar to the Z-Technique, one touch moves the object in

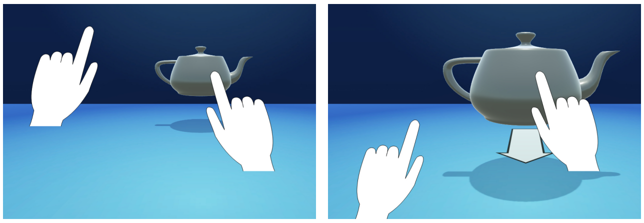


Figure 8: Changing object's depth with the Z-Technique [MCG10a].

the screen plane, and an indirect touch manipulates object depth. Two direct touches in the object enable rotations, using a constraint solver similar to Screen-Space [RDH09]. The authors compared DS3 with previous works [HtCC09,RDH09], and a user evaluation revealed that DOF separation led to better results. However, using a transformation plane parallel to the view plane can occasionally result in awkward transformations when the view plane is not orthogonal to one of the scene axes [MF11a].

Rather than using the number of users' touches to determine the type of transformation to apply, Liu et al. [LAFT12] use the movement characteristics of two touches. Two moving touches control 4 DOFs (3 translation and 1 rotation) in a manner similar to Sticky Fingers. One fixed touch and another moving touch control the remaining 2 DOFs. Although outperforming the screen-space and DS3 approaches and being comparable to Sticky Fingers while requiring less contact points, the authors state that their technique might not be very suitable for fine-tuning control of object transformations.

4.2. Indirect Interactions through Input Remapping

As we previously presented for mouse-based manipulations, a common approach for input remapping is the use of virtual widgets. Schmidt et al. [SSB08] introduced a 3D manipulation approach for sketch-based interfaces, combining 3D widgets, context-sensitive suggestions and gestural commands. Users indicate an object to transform by explicitly selecting it with a tap, and by drawing a stroke, the system responds by automatically creating translation and rotation widgets based on the candidate axis nearest to the stroke. Candidate axes include world and object axes. Initial widgets can be modified using context-sensitive gestures or by drawing a different axis.

To better understand user gestures for 3D manipulation tasks on multi-touch devices, Cohé et al. [CH12] conducted a user study and concluded that physically plausible interactions are favoured, and there are different strategies to develop an application focusing on broad usage or ease of use. Based on observations of users interacting with widgets for 3D manipulations, Cohé et al. [CDH11] designed a 3D transformation widget called tBox. This widget allows the direct and independent control of 9 DOFs (translation, rotation and scaling along each axis). tBox consists of a wire-frame cube, which is visible in Figure 9. Users can drag an edge of the cube to move the object in an axis containing the edge, and rotations are achieved by dragging one of the cube's faces.

To create VEs for computer-animated films, Kin et al. [KMB*11] designed and developed Eden, a fully functional multi-touch set construction application. Virtual objects can be translated in a horizontal plane using the usual direct drag approach and up and down with a second finger, similar to the Z-technique [MCG10a]. Rotations are performed similar to the Arcball [Sho92] widget. It also supports both uniform and one-dimensional scaling transformations.

LTouchIt [MLF11], although using direct manipulation for translations, also relies on widgets for rotations. Following the DOF separation, it has a set of interaction techniques that provide direct control of the object's position in no more than two simultaneous dimensions and rotations around one axis at a time using rotation handles. The translation plane is perpendicular to one of the scene axes and is defined by the camera orientation. Using the rotation handles, the user can select a handle to define a rotation axis and, with another touch, specify the rotation angle, as exemplified in Figure 10.

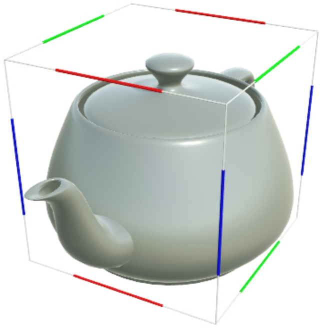


Figure 9: The tBox widget [CDH11].

Au et al. [ATF12] use the high input bandwidth of multi-touch surfaces and delegate the manipulation power of standard transformation widgets to multi-touch gestures. This enables seamless control of constraint and transformation manipulation using a single multi-touch action (Figure 11). Users can select a candidate axis with two touch points, and transforming the object is performed by holding and moving two fingers. This approach also supports plane constraints by using a candidate axis as the plane normal and transformations relative to a pivot point located on another object.

Regarding direct versus indirect interactions, Knoedel et al. [KH11] investigated the impact of the directness in TRS manipulation techniques. Their experiments indicated that a direct approach is better for completion time but that indirect interaction can improve both efficiency and precision.

Bollensdroff et al. [BHA12] redesigned older techniques for three-dimensional interactions [Hou92] using multi-touch input. They developed a cube-shaped widget, the Gimbal Box, which uses a touch in one of its faces to translate in the plane defined by that face. To rotate the object, the widget has two variations. One uses

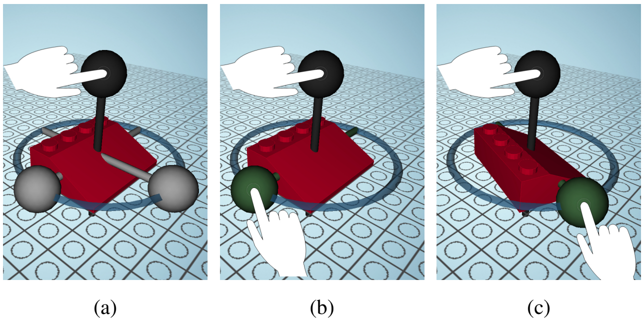


Figure 10: LTouchIt's rotation handles [MLF11]: selecting a rotation axis (a) and performing a rotation (b, c).

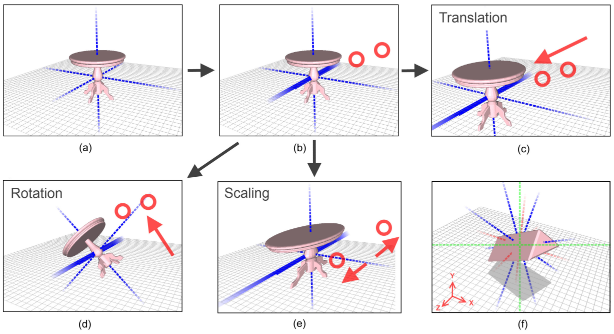


Figure 11: Multi-touch gestures for axis-based manipulations: (a) virtual object; (b) axis selection with two touches; (c)-(e) axis-constrained translation, rotation and scaling; (f) total set of candidate axes (extracted from [ATF12] with permission).

the TRS applied to a cube's face; alternatively, touching an edge of the box induces a rotation around an axis parallel to the edge. The other variation is based on Arcball [Sho92]. Through a controlled study, their techniques were compared to other approaches that are well known in the literature [HtCC09, RDH09]. They concluded that adapted widgets are superior to other approaches for multi-touch interactions, supporting DOF separation through the reduction of simultaneous control to 4 DOFs in a defined visible 2D subspace. Moreover, the authors suggest that "multi-touch is not the final answer" since "the projection of an object as input space for interaction can never reproduce precise motions of the object in 3D space".

TouchSketch [WCOM15], an interface for editing the shape of 3D objects, divides object manipulation into three categories: axis constrained, plane constrained and uniform manipulation. For this purpose, it resorts to a constraint menu, which allows users to select a constraint in the menu with the non-dominant hand and use the dominant hand to apply transformations respecting the selected constraint. Evaluation results suggest that this technique can outperform a single-touch approach based on widgets in terms of efficiency.

4.3. Touching Stereoscopic Tabletops

To improve both three-dimensional visualization and spatial perception, several researchers have explored interactions using stereoscopic environments. In such environments, since virtual objects can appear outside the surface, either in front of or behind the surface, previous touch techniques are not suitable. Directly touching the surface where the object is projected can disrupt the illusion and be unnatural, thus the need for different manipulation techniques. Considering the placement of virtual objects inside the tabletop in a fish-tank approach, touch solutions suffer from parallax issues [MZB12]. Above the table solutions have already been explored. Using the Responsive Workbench, one of the first stereoscopic tabletop VR devices, Cutler et al. [CFH97] constructed a system that allows users to manipulate virtual 3D models with both hands. The authors explored a variety of two-handed 3D tools and interactive techniques for model manipulation, constrained transformations and transitions between one- and two-handed interactions. However, they resorted to toolboxes to allow the user to transition between different operations.

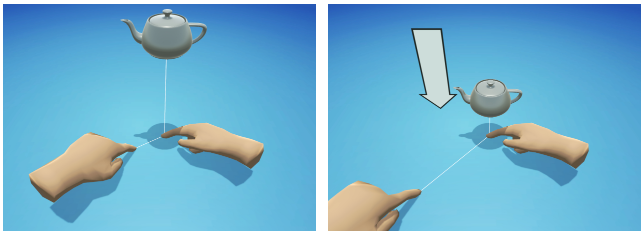
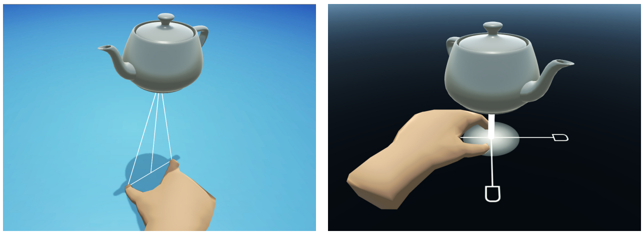


Figure 12: The balloon metaphor [BF07]: moving two fingers apart translates downwards.

Benko et al. [BF07] proposed a balloon metaphor to control a cursor (Figure 12), which is then used to manipulate three-dimensional virtual objects on a stereoscopic tabletop. Moving two fingers closer, the user allows the object to move up, and likewise, if the user moves the fingers away, the object will translate downwards. Later, Daiber et al. [DFK12] created a variation of this technique by adding a corkscrew metaphor, which can be used with either both hands or a single hand. With this approach, the user can use a circular motion in a widget rather than the distance between fingers to manipulate an object's height. The authors compared their technique with the previous techniques in both positive and negative parallax scenarios. Although none of the techniques was clearly identified as being better, the negative parallax space was shown to be more difficult to interact with.

Strothoff et al. [SVH11] proposed another approach to select and manipulate a cursor in stereoscopic tabletops. Using two fingers to define the base of a triangle, the height of the cursor, placed in the third vertex, is defined by the distance of the two touches, as exemplified in Figure 13b. Using this triangle cursor, users can manipulate selected objects in 4 DOFs: translation in three dimensions and rotations around a vertical axis.

To manipulate virtual objects in the full 9 DOFs, Toucheo [HBCdlR11] presented a setup with co-located 3D stereoscopic visualization, allowing people to use widgets on a multi-touch surface while avoiding occlusions caused by the hands. The authors combined a two-dimensional TRS interaction



(a) (b)

Figure 13: Object manipulation using: (a) the triangle cursor [SVH11], and (b) Toucheo's widgets [HBCdlR11].

on the surface with the balloon metaphor [BF07] and other widgets that provide both the remaining rotations and independent scaling along three axes (Figure 13a).

Previous works [BF07, DFK12, SVH11, HBCdlR11] prevent the vergence-accommodation conflict, which can lead to the loss of the stereoscopic effect or cause discomfort, by touching below the virtual object in the stereoscopic display. Simeone [Sim16] followed a different approach based on indirect touch interaction through an additional multi-touch surface. The author proposed two novel indirect manipulation techniques, Indirect4 and Indirect6, one to control 4 DOFs and the other for 6 DOFs, respectively. The first uses a touch from the dominant hand to control horizontal translations and a touch from the non-dominant hand to modify the object's vertical position (with vertical motions) and rotation around a vertical axis (with horizontal motions). The second technique manipulates the object's position similarly, but it uses two touches from the non-dominant hand to perform rotations. If the two fingers move horizontally or vertically, yaw or pitch is enabled, respectively. If they move in opposite directions, roll is enabled. These techniques were compared to DS3 [MCG10b] and Triangle Cursor [SVH11]. The results showed that indirect touch interaction techniques provide a more comfortable viewing experience while presenting no drawbacks when switching to indirect touch.

Giesler at al. [GVH14] proposed the Void Shadows technique for fish tank stereoscopic tabletops. This technique offers control over 4 DOFs (3 for translation and 1 for rotation) for each object present in the VE. Each object projects a fake shadow on the zero parallax plane, and the user is able to touch it directly. Direct translation from the finger position is applied to the object on the XY plane controlling 2 translation DOFs, while translation on the Z-axis is performed with a pinch gesture. The only rotation available is about the Z-axis and is performed by rotating two or more fingers in contact with the shadow around its centre. This technique allows all these interactions to be performed simultaneously if the user wishes to do so.

5. Mid-Air Interactions

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

In this way, interactions such as grab, move and rotate objects can be performed in immersive VEs, similarly to how they are performed with physical objects [RH92] (Figure 14). This Simple Virtual Hand manipulation [BKLJP04] is natural, but it can be challenging or not effective for some applications due to the limited range of translation and rotation and lack of precision [BMR12]. In Table 3 we summarize the techniques surveyed in this section.

5.1. Enabling technologies: handheld devices and hand trackers

Using inertial sensors, computer vision or magnetic tracking, the orientation and position of a handheld device can be derived and used for controlling virtual objects. Tracked handheld devices are the current solution proposed by the gaming VR industry with well-known commercial products such as Nintendo Wii, Playstation Move [psm17], HTC Vive [htc17], and Oculus Touch [ocu17]. These devices can provide at least 6-DOF tracking capabilities per controller, with extra DOFs depending on the number of buttons and control sticks that each controller possesses. In addition to being more suitable for video games and similar interactive applications with their button layout resembling those of standard gaming controllers or television remote controllers, they are also easier to track when compared to human body parts such as hands, limbs or heads.

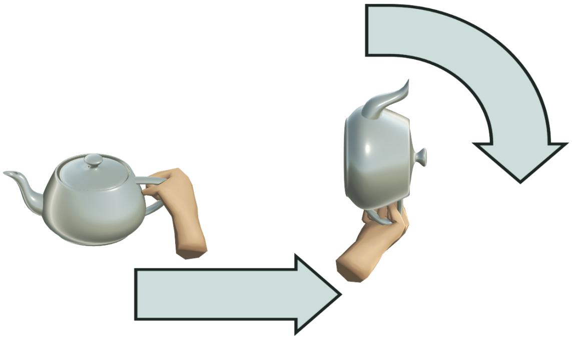


Figure 14: Simultaneous translation and rotation with the Simple Virtual Hand [BKLJP04].

Handheld manipulation devices can also be everyday life items, such as phones. In [KH10], it is shown that the 3-DOF orientation sensor of a phone can be effectively applied for controlling the orientation of a 3D virtual object.

Specific handheld devices have been designed for specific interaction and manipulation tasks. The Cube Mouse [FP00] is a 6-DOF tracked object with three rods that can be pulled and twisted, mapping other translational and rotational controls. This mouse was designed for specific visualization tasks supporting a bimanual control for moving and slicing objects.

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

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

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

More freedom than holding a device could be achieved using wearable devices (e.g. gloves [DSD08]). The Color Glove [WP09] enables precise finger and hand pose tracking. The system uses a simple RGB camera to capture the coloured areas of the gloves, being able to reconstruct the user's entire hand, thereby obtaining full 6-DOF tracking in real time. Manipulation based on wearable devices has been tested [CMD*14]; however, such tests showed that their use does not solve the usability issues of freehand manipulation, requiring the development of smart metaphors and feedback solutions. Furthermore, low-cost hardware solutions are not yet available.

A different approach in mid-air interaction is based on tracking hands without the need for handheld objects, exploiting depth sensors such as Microsoft Kinect or visible or IR stereo cameras (e.g. in the LeapMotion sensor). Wang et al. [WPP11] introduced a new way to track hands and fingers using affordable depth cameras. Their approach, in addition to pose detection, tracks each hand in 6 DOFs in a non-invasive manner. These tracking solutions allow hand reconstruction, which can be used to closely mimic physical interactions. The possibility of tracking fingers opens several possibilities [BMR12], but the tracking performances are not always satisfactory.

5.2. Mappings and metaphors

The greatest challenge for mid-air interactions is finding effective metaphors for mapping user movements in the 3D space to object movements, possibly exploiting existing motor programs and easy to perform user actions. Although the Simple Virtual Hand can handle the translation well, even if with some caveat for scaling, accuracy and release position, there are not easy solutions for rotation and scaling. For this reason, a vast amount of research has been dedicated to proposing effective solutions for motion mapping and manipulation metaphors.

Hilliges et al. [HIW*09] presented a technique to seamlessly switch between interactions on the tabletop and above it. The main goal of the authors was to create a solution that resembles physical manipulations, enabling depth-based interactions. Using computer vision, the user's hand is tracked in 4 DOFs (3 for translation and 1 for rotation), and the grab gesture can be detected. Shadows of the user's hands are projected into the scene, which are used to interact with virtual objects in three dimensions. After an object is grabbed by the user's shadow, the modifications in the corresponding hand are applied to the object.

Marquardt et al. [MJGJ11] also combined the multi-touch surface and the space above it in a continuous interaction space. Taking advantage of this space, they leveraged the user's hands movements to allow full 6-DOF interaction with digital content. Following this continuous space, Mockup Builder [ACJ12,ACJH13] offers a semi-immersive modelling environment in which users can freely manipulate three-dimensional virtual objects. The authors used GameTrak devices to follow the positions of users' hands in 3 DOFs, which acted as cursors, and adapted TRS to three dimensions to manipulate objects in mid-air with 7 DOFs (we will refer to this technique as Air-TRS [MFA*14]). With one hand users can directly grab and move an object, while a second hand, after performing a grab gesture outside the object, allows rotations around the first, as exemplified in Figure 15. Additionally, the distance between both hands is used for uniform scaling operations.

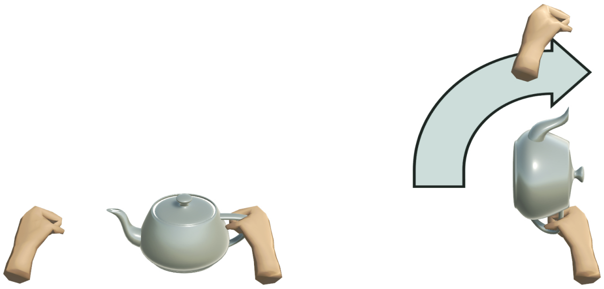


Figure 15: Rotation with the Air-TRS technique, as implemented in Mockup Builder [ACJH13].

Hilliges et al. [HKI*12] created a setup similar to Toucheo [HBCdlR11], the HoloDesk. It allows direct interaction with 3D graphics using a physical simulation and a depth camera for hand tracking.

Kim and Park [KP14] proposed a Virtual Handle with a Grabbing Metaphor (VHGM). When the user selects an object, the system generates a bounding sphere around the object. From the sphere's centre, a ray with its direction opposite to that of the virtual handle is projected to find the intersecting point on the sphere. This point serves as the reference frame for the following transformations (translation and rotation). User evaluation results suggest that VHGM can lead to better rotation efficiency than a standard 3D cursor.

Mapes and Moshell [MM95] introduced Spindle, a bi-manual technique to manipulate virtual objects. The point between user's hands is used to select the object and acts as the transformation center. Moving both hands at the same time in the same direction makes the object to translate, and moving them around the center rotates the object accordingly. Changing the distance between both hands scales the object. While Mapes and Moshell [MM95] used specific gloves as input devices, Bettio et al. [BGG*07] latter implemented this technique using two tracking cameras, and allowing a hands-free interaction. The effectiveness of this latter method was demonstrated with a simple application for model manipulation on a large stereo display, in which rendering constraints are met by employing state-of-the-art multiresolution techniques.

Song et al. [SGH*12] proposed a Handlebar metaphor (Figure 16), an approach similar to Spindle, using a single depth camera to track the position of users' hands in space. Since users' hands are only tracked in 3 DOF, rotations around the axis of the line defined by both hands can be achieved with an isolated swivel gesture. This technique also allows users to manipulate single objects or pack multiple objects along the handlebar. More recently, Cho et al. [CW15] proposed Spindle+Wheel, also based on Spindle [MM95] and similar to the Handlebar [SGH*12], developed for

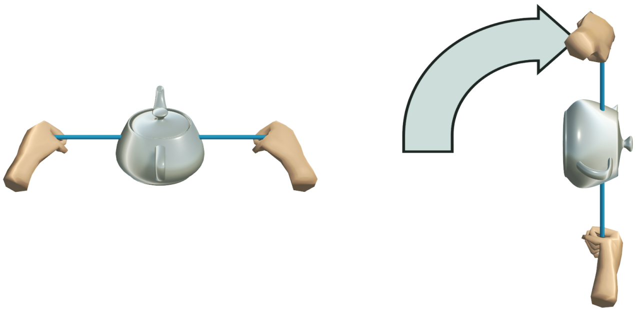


Figure 16: The Handlebar metaphor [SGH*12] being used to rotate a virtual object.

semi-immersive environments and resorting to spherical handheld devices for hand tracking. This approach uses an offset between users' hands and virtual cursors, and it is also two handed. Moving both hands in the same directions translates the object between them, and moving both hands in different directions rotates the object (roll and yaw). Changing the distance between hands performs scaling operations, and rotating one of the hands rotates around the main axis of the handheld device (pitch). The main difference from the Handlebar technique is that Spindle+Wheel offers simultaneous 7-DOF transformations.

Bossavit et al. [BMA*14] proposed two manipulation techniques: the Crank Handle (CH) and the Grasping Object (GO). The CH is a one-hand technique that separates translation from rotation and decomposes rotations in primary axes. These axes are selected through a crank handle metaphor. The GO technique is another one-hand manipulation technique. In contrast to the CH, it combines translation and rotation and does not decompose rotation in the primary axes. The authors based this technique on the RNT algorithm and on its 3D extension for 2-DOF inputs and extended it further to support 3-DOF positional input.

5.3. Mobile-device-based mappings and metaphors

When a tracked object is available, mappings can be enhanced by the use of objects' orientations and specific input channels.

Berge et al. [BDR15] proposed a classification for this specific manipulation technique that uses a common smartphone as a smart object to interact with a VE. The classification is based on three categories: around the smartphone (ASP), with the smartphone (WSP) and on the smartphone (OSP). Whereas the OSP category simply includes all the mere implementations of touch-based techniques on the phone screen with the sole difference of the effect of the interaction occurring remotely, the ASP and WSP categories offer a tool to classify techniques with an emphasis on the role played by the smartphone in the interaction. In WSP techniques, the smartphone is a traditional smart object used directly as a reference by the system to track the user's hand movement. Meanwhile, ASP techniques use the smartphone position as a reference frame for the dominant hand and its screen to provide visual feedback for the user.

Issartel et al. [IGIA16] analysed the manipulation of virtual objects through the use of a mobile device and the way the movement is mapped between the two. Three categories are presented: absolute position control, relative position control and rate control. The work offers insights on the benefits and disadvantages of the different solutions along with a more in-depth study on the implications caused by factors such as spatial feedback compliance and allocentric/egocentric design choices.

Speicher et al. [SDGK16] implemented a combined technique using both a Microsoft Kinect and a mobile phone to manipulate virtual objects. The Microsoft Kinect was used to track 3 DOFs for the hand position, while the mobile phone held in the user's dominant hand was used to track 3 DOFs for the rotation. The interaction technique was validated with a docking task and with measurements of the task completion time, translation task precision and rotation task precision. These last two measures were further subdivided by taking into account performances on the three different axes.

Mine et al. converted the desktop application SketchUp into a virtual reality application: VR SketchUp [MYC14, MYC15]. Their objective was to develop interaction techniques that can run across a spectrum of displays, ranging from the desktop and head-mounted displays to large CAVE environments, minimizing energy while maximizing comfort. For this purpose, they constructed a hybrid controller that collocates a touch display and physical buttons through a 6-DOF tracked smartphone attached to a handheld controller. 3D spatial input was used to achieve a coarse starting step. Meanwhile, 2D touch was used for precision input, such as controlling widgets, defining constraints and specific values for transformations, and providing numeric or textual input. To manipulate objects, the authors presented three alternatives: direct 6-DOF manipulation, where scaling of the object can be achieved using bi-manual interaction and DOF constraints, rotational axes, and special behaviour such as position-only manipulation are specified using the touchscreen interface; image plane interactions, where movement of the user's hand within their field of view is mapped to screen space interactions; and trackpad interaction, where the user manipulates objects via a touchpad widget on the touch screen to emulate mouse interactions within the user's screen space. Although the authors focused on several types of displays, resorting to imagery on the smartphone screen may not work well in conjunction with HMDs. However, some interactions on the touch surface were designed to not require the user to have to look down at them, such as menu navigation, which is represented by floating graphical elements in the VE.

5.4. Out-of-reach issues in immersive VEs

One of the first challenges addressed concerning the manipulation of objects in immersive VEs was how to extend users' capabilities by allowing interactions with objects that are out of reach of users' hands. The Go-Go immersive interaction technique [PBWI96] uses the metaphor of interactively growing the user's arm and nonlinear mapping for reaching and manipulating distant objects. When the user's hand moves above a certain distance, the arm grows according to a predefined coefficient. Below that distance, a 1:1 mapping

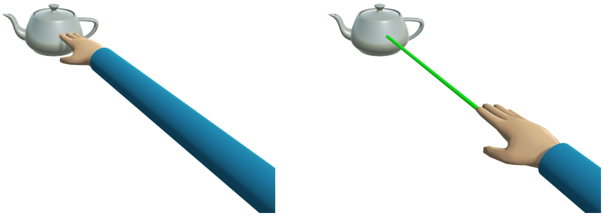


Figure 17: The Go-Go (left) and ray-casting (right) techniques to interact with out-of-reach objects [BH97].

is used. This technique allows for seamless direct manipulation of both nearby objects and those at a distance.

However, when comparing the Go-Go technique with other approaches (Figure 17), such as Stretch Go-Go, which improves on Go-Go by extending the virtual arm until infinity, and ray-casting, there is no clear winner [BH97]. User evaluation results showed significant drawbacks in all techniques. From this evaluation, HOMER (hand-centered object manipulation extending ray-casting) was proposed. It uses ray-casting to select the object, and after selecting the object, it moves the virtual hand to the object. The current distance between the user's body and hand is mapped to the distance to the virtual object. Therefore, manipulation is performed similarly to the Go-Go technique, but the scaling coefficient is calculated for each selected object.

A different approach for interacting with out-of-reach objects in large VEs is the Worlds in Miniature technique [SCP95]. Users can interact with a miniature of the virtual world to promptly move around and change their point of view or to manipulate virtual objects. Focusing on manipulating objects regardless of their scale, Pierce et al. [PSP99] proposed the Voodoo Dolls technique, which dynamically creates scaled handheld copies of the objects (dolls) that can be manipulated rather than the objects themselves. These dolls are used in pairs, one in each hand, and their effect depends on whether they are held in the right or left hand: the right hand's object is positioned in relation to the left hand's object. With this technique, users can work at multiple scales without explicitly resizing objects or the world.

5.5. Solving precision issues

To overcome the lack of precision with object positioning techniques in immersive VEs, Kiyokawa et al. [KTY97] proposed manipulation aids consisting of discrete placement constraints (snapping) and collision avoidance mechanisms. Without imposing placement restrictions, Frees et al. [FK05] introduced the PRISM (precise and rapid interaction through scaled manipulation) technique. In contrast to techniques such as Go-Go, which scale up hand movement to allow long-distance manipulation, PRISM scales the hand movement down to increase precision. Switching between precise and direct modes occurs according to the current velocity of the user's hand, as exemplified in Figure 18. When moving an object from one general location 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 and focus more on being precise. PRISM increases the control/display ratio, which causes the cursor or object to move more slowly than the user's hand, thereby reducing the effect of hand instability and creating an offset between the object and the hand. Using PRISM, the user is always in complete control of the position of the object being manipulated (in contrast to gravity and snapping techniques). User evaluation results show faster performance and higher user preference for PRISM over a traditional direct approach.

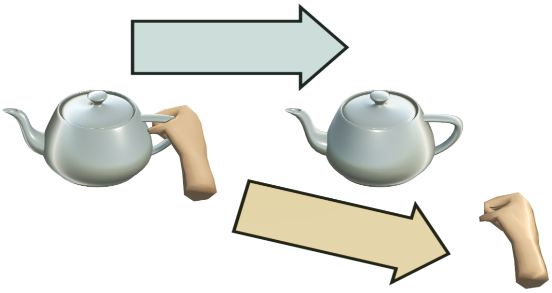


Figure 18: PRISM technique [FK05]. User moves the hand slowly to the right and down, while some horizontal movement and almost all vertical movement are scaled down.

The authors later extended the previous work by adding support for object rotation, which uses the angular speed of the hand [FKK07] and which the authors concluded to be confusing to users. They also presented how their approach can be useful for faster object selection using a 3D cursor, either for out-of-reach objects using a smoothed ray-casting approach or for cluttered environments, such as the Worlds in Miniature approach [SCP95].

Combining PRISM with the ray-casting-based approach HOMER [BH97], Wilkes et al. proposed Scaled HOMER [WB08], which uses velocity-based scaling to allow more precise manipulation at both near and far distances. It improved performance over HOMER in a wide variety of task conditions, primarily in those that require a high level of precision, object placement at a distance, or a large movement distance. Following Go-Go and PRISM studies, Auteri et al. [AGF13] combined both techniques to increase precision for extended reach 3D manipulation. The solution starts by applying PRISM to the movement of the user's hand (base cursor) directly, which calculates a new cursor position (PRISM cursor) based on velocity-based scaling. Then, the distance that the PRISM cursor moved is amplified by the Go-Go distance-based heuristic. The combination of Go-Go and PRISM provided a number of improvements, particularly task completion success and fine-grained manipulation.

One- and two-handed control techniques for precise positioning of 3D virtual objects in immersive VEs were proposed by Noritaka Osawa [Osa08]. This author proposed a position adjustment that consists of a scale factor for slowing hand movement, similar to PRISM [FK05], and a viewpoint adjustment that automatically approaches the viewpoint to the grabbed point such that the object being manipulated appears larger. To control the adjustments, two techniques are presented. The first uses only one hand and is based on its speed on the assumption that the user moves their hand slowly when they want to precisely manipulate an object. The other uses the distance between both hands. When the distance between them is small, the adjustments are activated. Through a user evaluation, the position and viewpoint adjustment methods showed improvements for small targets over a base scenario where this adjustments were disabled. Additionally, their results also showed that the two-handed control technique performed better than the one-handed technique.

Aguerreche et al. [ADL09] introduced a 3D interaction technique called 3-Hand Manipulation for multi-user collaborative manipulation of 3D objects. The 3-Hand Manipulation technique relies on the use of three manipulation points that can be used simultaneously by three different hands of two or three users. The three translation motions of the manipulation points can fully determine the resulting 6-DOF motion of the manipulated object. When a hand is close enough to the object to manipulate, ray-casting from the hand provides an intersection point with the object. This point is called a manipulation point. A rubber band is drawn between a hand and its manipulation point to avoid ambiguity concerning its owner and to display the distance between the hand and the manipulation point. It is elastic, and its colour varies according to the distance between the hand and the manipulation point. The authors indicate that a possible solution for implementing their technique is to use three point-to-point constraints of a physics engine.

Inspired by the previous work, Nguyen et al. [ND13] proposed a widget consisting of four manipulation points attached to objects, called the 3-Point++ tool, which includes three handle points, forming a triangle, and their barycentre. With this widget, users can control and adjust the position of objects. By moving the manipulation points, the position and the orientation of the object are controlled. The barycentre can be used for approximate positioning to control the object directly without constraints, while the three handle points are used for precise positioning. For this purpose, the barycentre has 6 DOFs, while the three handle points have only 3 DOFs. If one handle point is manipulated, then the object is rotated around an axis created by the two other handle points. If two handle points are manipulated at the same time, then the object is rotated around the third handle point. An evaluation was conducted comparing the 3-Point++ tool with a well-known technique using a 3D cursor to control an object directly with 6 DOFs. The 3-Point++ technique had the worst results due to its complexity.

Extending their previous work, Nguyen et al. [NDP14] presented the 7-Handle manipulation technique. This technique consists of a triangle-shaped widget with seven points, as depicted in Figure 19. Three points called first-level handles are the three vertices of the triangle, which act similarly to the 3-Point++ tool. The second-level handles are positioned at the midpoints of the three sides of the triangle and are used to control its two adjacent first-level handles. The last point, the third-level handle, is positioned at the centroid of the three first-level handles and can be used as a direct manipulation tool with 6 DOFs. The results of a user evaluation showed that the 7-Handle technique is only better suited than the traditional direct 6-DOF approach for manipulating large objects (side larger than 1.5 metres).

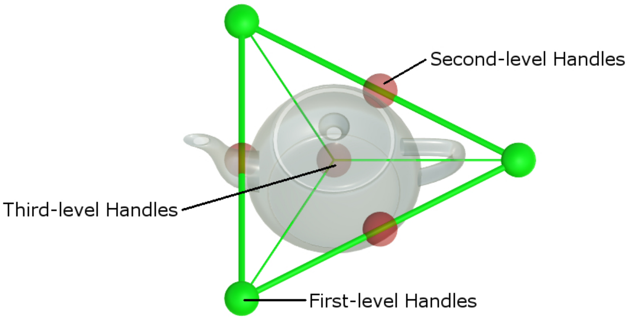


Figure 19: The 7-Handle tool [NDP14].

5.6. Analyses and comparisons

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

In [BIB*09], the use of 3D input is questioned after a comparison between mid-air manipulation with devices tracked in 6DOF and mouse-based methods on a placement task. Their experiment showed that, even with less DOF, the mouse was more efficient than the other devices. Its accuracy compensated the need to decompose tasks and it induced lower levels of stress.

Veit et al. [VCB09] studied the influence of the integration and separation of DOFs in orientation tasks in semi-immersive VEs. For this purpose, they compared an indirect mid-air technique (IR - indirect rotation), in which users can grab a virtual manipulator (a cube) and orient it by rotating the hand, with another where users manipulate each rotation axis independently in a multi-touch surface (BPCR - bi-manual plane-constrained rotations). Using the IR technique, users are able to combine three axes of rotation into a single gesture. With the BCPR technique, a 3-DOF orientation task can be decomposed into three 1-DOF sub-tasks by manipulating one axis at a time. User evaluation results showed that participants were faster with BPCR and revealed that even when using IR, participants tended to decompose tasks.

Schultheis et al. [SJT*12] performed a comparison between mouse, wand and a two-handed interface for 3D virtual object and world manipulation through user evaluation, using both monoscopic and stereoscopic displays (although no discussion is provided regarding viewpoint correlation or co-location of users' hands and virtual imagery). The mouse interface resorted to manipulators (or widgets) for controlling translation and rotation angles for each axis. The wand behaved as a regular 6-DOF tracked device, allowing direct manipulation of the selected object. The two-handed approach is very similar to the Handlebar [SGH*12]. The two-handed interface out-performed the mouse and wand, and the wand out-performed the mouse, albeit requiring appropriate training. The authors stated that these results suggest that well-designed many-DOF interfaces have an inherent advantage over 2-DOF input for fundamental 3D tasks.

Vuibert et al. [VSC15] compared the performance of three mid-air interaction options using either a physical replica of the virtual object, a wand-like device or the user's fingers. For this purpose, they conducted a user evaluation with a docking task with 6 DOFs. As a baseline, they resorted to a mechanically constrained input device, the Phantom Omni. The authors found that the Phantom was the most accurate device for position and orientation, whereas the tangible mid-air interactions (wand and object's replica) were the fastest. Although the fingers did not outperform the Phantom in accuracy or speed, the difference between these two conditions was small. Moreover, subjects preferred the wand and fingers, while interaction with the replica was the least favoured.

Mendes et al. performed a comparative study between different interactions for 3D object manipulations using a setup that combines spatial 6-DOF Hand tracking and a multi-touch stereo tabletop [MFA*14]. The authors compared a touch approach similar to Toucheo [HBCdlR11] and four mid-air techniques: 6-DOF Hand, a direct approach that uses the dominant hand to grab, move and rotate objects and the distance between both hands for scale; 3-DOF Hand, in which the dominant hand only moves the object, while rotation and scaling are given by the non-dominant hand; Air-TRS, as used in Mockup Builder [ACJH13]; and the Handlebar [SGH*12]. User evaluation results suggest that mid-air interactions are better than touch based, and 6-DOF Hand and Handlebar are both faster and preferred by participants.

Caputo and Giachetti [CG15] conducted an evaluation and comparison of four mid-air manipulation techniques using low-cost hand tracking sensors. The examined techniques ranging from direct to more indirect metaphors to study the effectiveness of DOF separation and hybrid solutions for different manipulation actions, such as translation and rotation. The usability of the methods was tested in an immersive VR environment with test subjects performing a simple docking task (Figure 20). The results showed better performance for all the techniques using a more indirect approach for rotation actions.

Feng et al. [FCW15] conducted an evaluation similar to Mendes et al. [MFA*14], but they used a different setup with held devices. Rather than co-locating users' hands with stereoscopic imagery, they used a fish tank stereoscopic visualization with offset manipulation techniques. Similarities in the results with the previous study lead to a tentative guideline: if satisfying each individual user's preference is of high importance to the interface designer, provide the user the option of Spindle+Wheel (Handlebar) or Grab-and-Scale (6-DOF Hand) derived methods; otherwise, use Grab-and-Scale (6-DOF Hand).

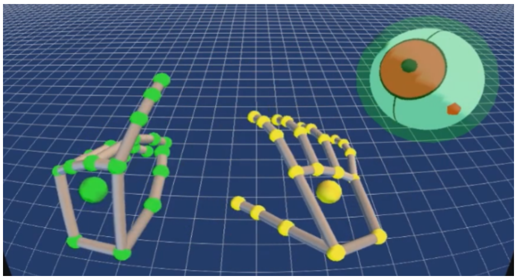
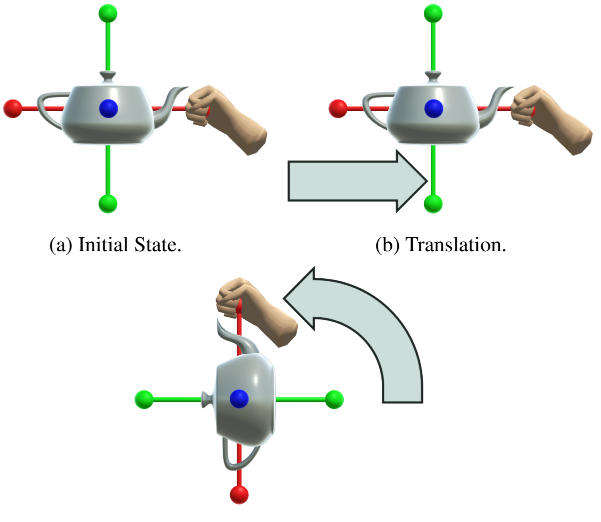


Figure 20: Snapshot of the docking task used to evaluate the techniques from [CG15].

Moehring et al. [MF11b] presented a study that compares finger-based interaction to controller-based interaction in a CAVE and in a HMD for exploration of car models. The authors focused on interaction tasks within reach of the users' arms and hands and explored several feedback methods, including visual, pressure-based tactile and vibrotactile feedback. The results suggest that controller-based interaction is often faster and more robust since the button-based selection provides very clear feedback on the start, stop and status of the interaction. However, finger-based interaction is preferred over controller-based interaction for the assessments of various functionalities in a car interior, as the abstract character of indirect metaphors leads to a loss of realism and therefore impairs the judgement of the car interior. Grasping feedback is a requirement to judge grasp status. It is not sufficient to simply have an object follow the user's hand motion once it is grasped. Although visual feedback alone is mostly sufficient for HMD applications, tactile feedback significantly improves interaction independent of the display system. Vibrational feedback is considerably stronger than pressure-based sensations but can quickly become annoying.

Motivated by the results obtained by DOF separation in mouse- and touch-based manipulation techniques, Mendes et al. [MRFJ16] assessed its impact on spatial interactions in IVEs. In this study, an approach based on virtual handles [CSH*92] (Figure 21) that restrict all transformations to a single DOF was evaluated. As baseline, a Simple Virtual Hand [BKLJP04] and PRISM [FKK07] were used. The results showed that DOF separation through virtual widgets can lead to error reduction at the cost of increased time for complex tasks. From this result, a set of developmental guidelines was proposed: direct manipulation is well suited for coarse transformations; translation and rotation operations should be separated whenever possible to prevent unwanted transformations; single DOF separation is very desirable for precise transformations,



(a) Initial State.

(b) Translation.

(c) Rotation.

Figure 21: Widgets technique based on virtual handles to translate and rotate objects [MRFJ16].

typically for fine-grain adjustments; and scaled transformations, as proposed in PRISM, are appealing only for translation, as scaled rotation confused participants.

6. Discussion

Our analysis is focused on 9-DOF manipulation, including rotation, translation and scaling. We have seen that a variety of methods have been presented, each with its own features. Of course, to derive useful guidelines for design and to understand potential future research efforts, it is necessary to understand the different constraints of real-world applications in terms of display immersion and the types and performances of input devices.

6.1. Trends in mouse- and keyboard-based interactions

As shown in Section 3, methods for 3D manipulation in desktop environments currently appear to be well established. The main editors and applications for 3D design generally exploit similar techniques and widgets, derived from techniques that are more than two decades old. Table 1 summarizes the surveyed works in 3D desktop manipulations. Naturally, and characteristic to desktop environments, all displayed imagery is screen constrained, and the tracking space is 2D separated. All allow single DOF control, with the exception of Two Pointer. While the former became the default for mouse-based manipulations, the latter became the basis for multi-touch interactions with two contact points, primarily for 2D manipulations, including the rubber band metaphor for scaling operations.

6.2. Design guidelines and considerations for desktop 3D interfaces

Analysing existing mouse- and keyboard-based interfaces to manipulate 3D virtual objects, the following considerations can be identified:

- The main challenge in desktop 3D virtual object manipulation is the mapping of the 2D input to 3D transformations. To overcome this challenge, either a multiple viewport approach and/or specific widgets are generally used.
- The multiple viewport approach uses different views of the virtual scene. These are orthogonal projections, and their view vector is coincident to a scene axis. Consequently, all interactions can be restricted to a single plane for each viewport, taking advantage of simpler 2D interactions and a more direct mapping between input and output.
- Widgets are a common alternative that allows interactions with unconstrained perspective projections of the 3D virtual environment. These consist of additional virtual objects that allow users to explicitly select specific transformations and axes to be applied onto the desired object.
- Using multi-DOF devices rather than a mouse [BIB*09] does not provide measurable advantages and may be recommended only for specific applications or user categories.

6.3. Trends in touch-based manipulation

The main features of touch-based 3D manipulation interfaces proposed in the literature are summarized in Table 2. From analyzing this table, we observe that most techniques are conceived for co-located environments, as expected for multi-touch interfaces. It is also possible to observe that most perform DOF separation, decoupling not only transformations but also DOFs in each transformation supported. However, few explore scaling operations, and there are many remappings due to the dimensionality disparity between input and output.

Although 2D interaction has found easy-to-use *de facto* standards for multi-touch devices, adapting these standards to manipulate 3D objects is not trivial in that it requires mapping 2D input spaces to 3D virtual worlds. However, the devices allow users to directly touch the objects displayed, providing consistent feedback.

Attempting to create more natural interactions, researchers initially proposed techniques for controlling several DOFs at the same time [HtCC09, RDH09]. Nonetheless, reduction in the number of DOFs simultaneously controlled has been suggested [MCG10a, MCG10b] and followed by several authors. Thus, techniques that allow manipulations with many DOFs but with few controlled simultaneously and totally separating transformations have been proposed. Similar to mouse-based manipulations, researchers turned to virtual widgets to clearly and unambiguously select the transformation and axis [CDH11,MLF11,BHA12]. Indeed, evaluation results suggest that those improve users' performance. Even when interacting with stereoscopic imagery above tabletops, the only technique that allows full 9-DOF manipulations [HBCdlR11] resorts to widgets.

6.4. Design guidelines and considerations for touch-based interfaces

As a take-home message from emerging trends and literature comparisons, we can derive some useful guidelines and considerations.

- The great advantage of touch-based interfaces is the possibility to interact with virtual content by directly touching it with ones' fingertips. This allows for more natural interactions, as physical manipulation metaphors can be employed, potentially reducing techniques' learning curves.
- When considering direct approaches that follow exact mappings, it has been shown that the numbers of input and output DOFs should be close. Thus, higher DOF transformations should be associated with a higher number of contact points.
- The main issue with direct touch approaches for object manipulation is that, when controlling multiple DOFs simultaneously, unwanted transformations occur. To prevent this, DOF separation for touch interactions has been suggested. By manipulating fewer DOFs at each moment, users have increased control over the outcome, which can also increase the efficiency of the manipulations.
- DOF separation can consist of both separating different transformations and restricting a transformation to specific axes. For separating transformations, a common way to achieve this is by using a different number of touches for each transformation

(e.g. one finger translates, two fingers rotate). To identify a single transformation axis, the vector defined by two fingers can be used. However, finding an adequate projection of such a 2D vector to the virtual scene to define a 3D vector might be challenging.

- To handle the 2D-3D remapping exploiting DOF separation, virtual widgets have been proven to be quite useful. Widgets can show all the manipulations available for an object or a set of transformations according to specific axis through user sketching. They ease the process of remembering how to perform restricted transformations, generally by touching on specific handles.
- When implementing techniques for specific scenarios or devices, the available interaction space should be taken into account. For instance, techniques that resort to multiple hands or fingers are generally better suited for large surfaces, such as tabletops and wall displays. The limited space of tablets and smartphones can complicate their usage, and the content can even be occluded by users' hands and fingers. In these cases, techniques with less contact points are better suited.
- When using touch to interact with stereoscopic imagery, different challenges arise since directly touching on a displayed object might disrupt depth's illusion and/or suffer from parallax issues. Proposed solutions follow indirect approaches, either by touching outside the object, typically resorting to some type of widget to remap users' actions, or by using separated interaction spaces through additional touch-enabled surfaces. However, an exhaustive evaluation on which approach is better suited for each type of scenario (positive and negative parallax) is still lacking.

6.5. Trends in mid-air interface design

Similar to touch-based techniques, we applied our taxonomy to classify techniques for mid-air manipulation, as reported in Table 3. From this table, we can conclude that most techniques, although being developed for several types of displays and tracking solutions, resort to exact mappings due to the naturality offered by spatial input. Thus, very few explore transformation separation, and only partially. Even less support DOF separation within a transformation type. Again, as in touch-based manipulations, few explore scaling operations, and those that do only support uniform scaling.

Having an input with higher DOFs as a direct mapping between input and output, most current mid-air approaches for 3D virtual object manipulation attempt to mimic physical world interactions [BKLJP04,HIW*09,ACJH13,SGH*12,WPP11], having no separation of transformations. Having realized that human accuracy is limited, occasionally aggravated by input devices' resolution, efforts have been conducted to alleviate this issue.

To improve manipulations' accuracy, authors have already attempted to either scale down hand motions [FK05, FKK07] or move the viewpoint closer to the object being manipulated [Osa08], but without regard to DOF separation. Indeed, almost no mid-air techniques with exact mappings even separate translation and rotation. The only exceptions are Air-TRS [ACJH13] and 3-DOF Hand [MFA*14], as transformations are enabled with different hands, being possible to translate without performing any rotation. However, they do not allow performing rotations with one hand without having the other hand being engaged in translations.

Conversely, approaches based on virtual widgets have been proposed [ND13, NDP14] to limit simultaneous transformations. However, these techniques do not provide promising results: 3-Point++ [ND13] performs worse than direct manipulation with 6 DOFs, and 7-Handle [NDP14] is only suited for very large objects. Other more familiar virtual widgets have recently been explored for mid-air [MRFJ16], using common reference frames and single DOF manipulation. This DOF separation leads to an increased accuracy but penalized task times.

Although most techniques consist of translation and rotation, scaling is often disregarded. Some touch approaches that resort to widgets allow for single DOF scaling [CDH11, ATF12, WCOM15, HBCdlR11]. However, in mid-air, techniques that offer scaling capabilities only perform uniform scaling [SGH*12, CW15, ACJH13, MFA*14].

6.6. Design guidelines and considerations for mid-air interfaces

From the analysis of design trends and system comparisons, we derived some useful insights.

- Exact mapping between tracked hand/device and virtual object has often been followed in mid-air interactions. This is the most natural approach as it mimics physical interactions, and studies have shown that it is well suited for coarse transformations.
- A result of exact mappings is that they may require additional movement of the user, either physical or virtual. When designing interactions for large environments, techniques that explore scaled mappings might be useful for extending users' reach.
- Although techniques that move away from direct manipulations are less natural, they can avoid unwanted side effects of replicating the physical world exactly, and they can provide users with enhanced abilities that may improve performance and usability [BMR12].
- Orientation control would benefit from smart remapping as the use of direct mapping on hand/device does not provide good results in general. Bi-manual solutions (e.g. Handlebar) appear to be the most natural ones and appeared in several research systems. However, they have, in general, some drawbacks making them not suitable for all applications: the necessity of using two hands, the necessity of splitting large rotations into sub-parts due to physiological constraints and the large motions performed that may be fatiguing and causing the "gorilla arm" effect [LaV17].
- Accuracy in mid-air manipulation is still a relevant issue. Finding a familiar manipulation metaphor that provides a satisfactory level of accuracy is an interesting open challenge for research.
- A possible approach to increase precision is to scale down users' hand motions. However, it has been shown that it is only appealing for translations. Scaled rotations generally confuse users, severely decreasing overall performance.
- DOF separation, achieved, for instance, through virtual widgets, is common in mouse- and touch-based solutions, and it is now starting to also be explored in mid-air, showing benefits in specific conditions. It can provide better accuracy and prevent unwanted transformations. However, in complex tasks, it requires

multiple selections and switching between different types of interaction modes, which might not be efficient and appreciated. Thus, DOF separation has to be further explored to be a viable alternative to more common interfaces. Context-specific trade-offs should be found and adapted to end-user applications.

- Although only translation and rotation are required for positioning tasks, scaling is often grouped together with those transformations in specialized software. However, none of the reviewed mid-air techniques offer full 9-DOF manipulations, as those that support scaling only do so uniformly. Techniques to perform free 1, 2 and 3 DOF scaling, existing in mouse- and touch-based interfaces, are worth exploring.
- VR setups are quite diverse in terms of input hardware constraints, and manipulation solutions must be adapted to the setup constraints. We expect that a vast amount of future applications will be based on emerging low-cost setups (e.g., Oculus Rift, HTC Vive, Playstation VR) with specific gesture capture systems and handheld devices, and the adaptability of the presented techniques to these types of setups will be a key factor of success.
- Whereas we can rely on standardized widgets and interaction habits on traditional displays and touchscreens, there is still a lack of well-accepted standards in 3D gestural interaction. If an effective and precise manipulation cannot be attained with natural mid-air gestures for some environments, it would be helpful for the emergence of standards in mid-air gestural interfaces to increase learnability.
- Interface design should be carefully adapted to the type of immersive display used. There are differences between HMDs and CAVEs or stereoscopic tabletops. A number of interface issues arise with stereo displays, as stated by Bowman et al. [BKLJP01]. For instance, because users can see their own hands in front of the display, they can inadvertently block out virtual objects that should appear to be closer than their hands. With HMDs, since users do not see the position and orientation of their bodies and limbs, solutions must be explored to increase users' proprioception [MBJS97].

6.7. Open Challenges

Mouse- and touch- based manipulations already have very mature research on how to manipulate virtual objects with different levels of control, with both multiple simultaneous DOF controlled and single DOF transformations to ease input mapping. For mid-air however, most techniques still follow direct mappings with a high number of DOF controlled at the same time, which are only suitable for coarse transformations. Further exploration of techniques to increase precision in mid-air manipulations following DOF separation could provide interesting contributions.

Besides developing techniques that either offer simultaneous control of multiple DOF or allow single DOF manipulation, a possibility for future work could be to explore techniques with adjustable DOF control, which could allow users to explicitly define multiple DOF from different transformations for simultaneously control. For instance, it could allow the specification of plane-like constraints, offering translation restricted to 2 DOF in mid-air instead to a single DOF, or using the 2D TRS approach in such plane allowing 2 DOF translation and 1 DOF rotation at the same time. This might be a good complement to transformation separation and single DOF manipulation, usually followed for DOF separation.

Out-of-reach manipulations also offer possibilities for future research. Existing approaches typically scale up user movements, leading to even more inaccurate manipulations as it amplifies hand and tracker jitter.

Research on all the relevant factors affecting the virtual manipulation experience in different contexts will also be useful. In this case, for example, a particular role is played by the spatial abilities of the user target, as mental rotation ability in particular can have an impact on the use of manipulation interfaces [BH14]. Additionally, selection and release strategies can also affect the performance of manipulations; thus, they need to be addressed when implementing manipulation techniques.

It is also to be considered that the technical evolution of tracking and vision technologies will significantly change the performances of the different methods presented, making them more suitable for manipulation control and changing the future research priorities. For example, finger-tracking-based techniques may now have a limited range of applications due to limitation in low-cost tracking reliability, but they could be more effective with the evolution of the related technologies. Low cost VR gloves that are reaching the market may increase the usability of finger based interfaces. Some of these devices will also provide haptic feedback features. Additional feedback cues can improve manipulations in mid-air, and the effectiveness of their use needs to be evaluated.

Other research efforts will be surely necessary to evaluate manipulation tools in mixed/augmented reality (AR) applications, that are expected to be widespread in the future (especially within the Industry 4.0 framework). Manipulation of virtual objects in real scenes may require specific design choices that need to be investigated.

In general, we believe that the availability of constantly improved viewers and hand tracking systems will enhance the search for new approaches for mid-air manipulations that are easy to learn and use. The availability of reliable mid-air interfaces is a key factor to properly take advantage of the ever more common visualizations that VR and AR can offer, as well as the spatial input often associated. It could make these technologies really useful for 3D content creation, and eventually render obsolete the traditional desktop setups and decades-old WIMP interfaces still common in a majority of fields.

7. Conclusions

This survey shows that a considerable amount of research effort has been dedicated to the challenge of allowing an easy and precise solution to 9-DOF rigid manipulation in different types of virtual environments and that several issues are hidden in this apparently simple task.

No generic solutions can be derived from the literature analysis, as manipulation methods have been proposed and tested for a variety of virtual environments, designed for different applications and with different constraints given by visualization and tracking systems. Considering this limitation, we were able to provide an analysis of research trends and to suggest a few design guidelines derived from the literature and adapted to different environments and system constraints. We also pointed out possible future directions for research on virtual object manipulation.

We believe that the outcomes of this survey can be useful for the development of effective applications, as a clear threat for the success of VR technologies is the lack of usability, and manipulation is one of the most critical tasks for this [LaV17]. Many VR applications and projects in the past failed due to unforeseen usability issues compromising the effectiveness of the interactive systems.

Examining the literature, it must also be noted that experimental validation is often limited due to the objective difficulty and cost of setting up carefully designed user tests. For these reasons, experiments and comparisons may occasionally provide apparently contradictory results, often due to different setups/conditions, low number of subjects, and many other factors. Standardization of specific user tests could mitigate this issue and better frame new contributions to the field.

Acknowledgements

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT), through grants UID/CEC/50021/2013, TECTON 3D (PTDC/EEI-SII/3154/2012), IT-MEDEX (PTDC/EEISII/6038/2014) and doctoral grant SFRH/BD/91372/2012, and by project MIUR Excellence Departments 2018-2022.

References

[3dc17] 3Dconnexion. www.3dconnexion.com, 2017. [Online; accessed 3-January-2018]. 6

[AA13] ARGELAGUET F., ANDUJAR C.: A survey of 3d object selection techniques for virtual environments. *Computers & Graphics 37*, 3 (2013), 121–136. 2, 3

[ACJ12] ARAÚJO B. R. D., CASIEZ G., JORGE J. A.: Mockup builder: direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of GI '12* (2012), Canadian Information Processing Society, pp. 173–180. 11

[ACJH13] ARAUJO B. R. D., CASIEZ G., JORGE J. A., HACHET M.: Mockup builder: 3d modeling on and above the surface. *Computers & Graphics 37*, 3 (2013), 165 – 178. doi:10.1016/j.cag.2012.12.005. 2, 11, 15, 17, 25

[ADL09] AGUERRECHE L., DUVAL T., LÉCUYER A.: 3-hand manipulation of virtual objects. In *Proceedings of the 15th Joint virtual reality Eurographics conference on Virtual Environments* (2009), Eurographics Association, pp. 153–156. 14

[AGF13] AUTERI C., GUERRA M., FREES S.: Increasing precision for extended reach 3d manipulation. *The International Journal of Virtual Reality 12*, 1 (2013), 66–73. 13

[AJC14] AZEVEDO A. S., JORGE J., CAMPOS P.: Combining eeg data with place and plausibility responses as an approach to measuring presence in outdoor virtual environments. *PRESENCE: Teleoperators and Virtual Environments 23*, 4 (2014), 354–368. 2

[ATF12] AU O. K.-C., TAI C.-L., FU H.: Multitouch gestures for constrained transformation of 3d objects. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 651–660. 8, 9, 17, 24

[BDR15] BERGÉ L.-P., DUBOIS E., RAYNAL M.: Design and evaluation of an "around the smartphone" technique for 3d manipulations on distant display. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (New York, NY, USA, 2015), SUI '15, ACM, pp. 69–78. doi:10.1145/2788940.2788941. 12

[BF07] BENKO H., FEINER S. K.: Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. In *3DUI* (2007), p. 22. 9, 10

[BGG*07] BETTIO F., GIACHETTI A., GOBBETTI E., MARTON F., PINTORE G.: A practical vision based approach to unencumbered direct spatial manipulation in virtual worlds. In *Eurographics Italian Chapter Conference* (2007), pp. 145–150. 11

[BH97] BOWMAN D. A., HODGES L. F.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (1997), ACM, pp. 35–ff. 13, 25

[BH99] BOWMAN D. A., HODGES L. F.: Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages & Computing 10*, 1 (1999), 37–53. 3, 4

[BH14] BARRETT T. J., HEGARTY M.: Interaction design and the role of spatial ability in moderating virtual molecule manipulation performance. In *CogSci* (2014). 18

[BHA12] BOLLENSDORFF B., HAHNE U., ALEXA M.: The effect of perspective projection in multi-touch 3d interaction. In *Proceedings of Graphics Interface 2012* (Toronto, Ont., Canada, Canada, 2012), GI '12, Canadian Information Processing Society, pp. 165–172. 8, 16, 24

[BIB*09] BÉRARD F., IP J., BENOVOY M., EL-SHIMY D., BLUM J. R., COOPERSTOCK J. R.: Did "minority report" get it wrong? superiority of the mouse over 3d input devices in a 3d placement task. In *IFIP Conference on Human-Computer Interaction* (2009), Springer, pp. 400–414. 14, 16

[Bie90] BIER E. A.: Snap-dragging in three dimensions. *ACM SIGGRAPH Computer Graphics 24*, 2 (1990), 193–204. 3

[BKLJP01] BOWMAN D. A., KRUIJFF E., LAVIOLA JR J. J., POUPYREV I.: An introduction to 3-d user interface design. *Presence: Teleoperators and virtual environments 10*, 1 (2001), 96–108. 2, 18

[BKLJP04] BOWMAN D., KRUIJFF E., LAVIOLA JR J. J., POUPYREV I.: *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004. 2, 3, 10, 15, 17, 25

[BM*07] BOWMAN D., MCMAHAN R. P., ET AL.: Virtual reality: how much immersion is enough? *Computer 40*, 7 (2007), 36–43. 2

[BMA*14] BOSSAVIT B., MARZO A., ARDAIZ O., DE CERIO L. D., PINA A.: Design choices and their implications for 3d mid-air manipulation techniques. *Presence: Teleoper. Virtual Environ. 23*, 4 (Nov. 2014), 377–392. doi:10.1162/PRES_a_00207. 12, 25

[BMR12] BOWMAN D. A., MCMAHAN R. P., RAGAN E. D.: Questioning naturalism in 3d user interfaces. *Communications of the ACM 55*, 9 (2012), 78–88. 10, 11, 17

[Bol80] BOLT R. A.: *"Put-that-there": Voice and gesture at the graphics interface*, vol. 14. ACM, 1980. 3

[CDH11] COHÉ A., DÈCLE F., HACHET M.: tbox: a 3d transformation widget designed for touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 3005–3008. doi:10.1145/1978942.1979387. 8, 16, 17, 24

[CFH97] CUTLER L. D., FRÖHLICH B., HANRAHAN P.: Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (1997), ACM, pp. 107–114. 9

[CG15] CAPUTO F. M., GIACHETTI A.: Evaluation of basic object manipulation modes for low-cost immersive virtual reality. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter* (2015), ACM, pp. 74–77. 15

[CH12] Cohé A., Hachet M.: Understanding user gestures for manipulating 3D objects from touchscreen inputs. In *Graphics Interface* (Toronto, Canada, May 2012), ACM, pp. 157–164. 8

[CMD*14] Chapoulie E., Marchal M., Dimara E., Roussou M., Lombardo J.-C., Drettakis G.: Evaluation of direct manipulation using finger tracking for complex tasks in an immersive cube. *Virtual Reality 18*, 3 (2014), 203–217. 11

[CSH*92] Conner B. D., Snibbe S. S., Herndon K. P., Robbins D. C., Zeleznik R. C., Van Dam A.: Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics* (1992), ACM, pp. 183–188. 5, 6, 15, 23

[CW15] Cho I., Wartell Z.: Evaluation of a bimanual simultaneous 7dof interaction technique in virtual environments. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on* (2015), IEEE, pp. 133–136. 11, 17, 25

[DBB15] Davis B. A., Bryla K., Benton P. A.: *Oculus Rift in action*. Manning, 2015. 2

[DFK12] Daiber F., Falk E., Krüger A.: Balloon selection revisited: multi-touch selection techniques for stereoscopic data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2012), AVI '12, ACM, pp. 441–444. doi:10.1145/2254556.2254641. 9, 10

[DSD08] Dipietro L., Sabatini A. M., Dario P.: A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38*, 4 (2008), 461–482. 11

[FCW15] Feng J., Cho I., Wartell Z.: Comparison of device-based, one and two-handed 7dof manipulation techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (2015), ACM, pp. 2–9. 15

[FHSH06] Froehlich B., Hochstrate J., Skuk V., Huckauf A.: The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 191–199. 6

[FK05] Frees S., Kessler G. D.: Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE* (2005), IEEE, pp. 99–106. 13, 17

[FKK07] Frees S., Kessler G. D., Kay E.: Prism interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI) 14*, 1 (2007), 2. 13, 15, 17, 25

[FP00] Fröhlich B., Plate J.: The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2000), ACM, pp. 526–531. 10

[Fro05] Froehlich B.: The quest for intuitive 3d input devices. *HCI International, Las Vegas, Nevada USA* (2005). 4

[GMM*14] Guerreiro J., Medeiros D., Mendes D., Sousa M., Jorge J., Raposo A., Santos I.: Beyond post-it: structured multimedia annotations for collaborative ves. In *Proc. of the Eurographics Conference on Virtual Environments* (2014), pp. 55–62. 2

[GVH14] Giesler A., Valkov D., Hinrichs K.: Void shadows: Multi-touch interaction with stereoscopic objects on the tabletop. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction* (New York, NY, USA, 2014), SUI '14, ACM, pp. 104–112. doi:10.1145/2659766.2659779. 10, 24

[GW07] Grossman T., Wigdor D.: Going deeper: a taxonomy of 3d on the tabletop. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on* (2007), IEEE, pp. 137–144. 2

[Han05] Han J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), ACM, pp. 115–118. 7

[HBCdlR11] Hachet M., Bossavit B., Cohé A., de la Rivière J.-B.: Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2011), UIST '11, ACM, pp. 587–592. doi:10.1145/2047196.2047273. 9, 10, 11, 15, 16, 17, 24

[HCC07] Hancock M., Carpendale S., Cockburn A.: Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 1147–1156. doi:10.1145/1240624.1240798. 7

[HGR03] Hachet M., Guitton P., Reuter P.: The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2003), ACM, pp. 225–112. 6

[HIW*09] Hilliges O., Izadi S., Wilson A. D., Hodges S., Garcia-Mendoza A., Butz A.: Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (New York, NY, USA, 2009), UIST '09, ACM, pp. 139–148. doi:10.1145/1622176.1622203. 11, 17, 25

[HKI*12] Hilliges O., Kim D., Izadi S., Weiss M., Wilson A.: Holodesk: direct 3d interactions with a situated see-through display. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 2421–2430. doi:10.1145/2208276.2208405. 4, 11

[Hou92] Houde S.: Iterative design of an interface for easy 3-d direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1992), CHI '92, ACM, pp. 135–142. doi:10.1145/142750.142772. 5, 8, 23

[htc17] HTC Vive. www.vive.com, 2017. [Online; accessed 3-January-2018]. 10

[HtCC09] Hancock M., ten Cate T., Carpendale S.: Sticky tools: full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2009), ITS '09, ACM, pp. 133–140. doi:10.1145/1731903.1731930. 7, 8, 9, 16, 24

[HVW*06] Hancock M., Vernier F., Wigdor D., Carpendale S., Shen C.: Rotation and translation mechanisms for tabletop interaction. In *Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on* (2006), pp. 8 pp.–. doi:10.1109/TABLETOP.2006.26. 7

[HZS*13] Hughes C. E., Zhang L., Schulze J. P., Edelstein E., Macagno E.: Cavecad: Architectural design in the cave. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on* (2013), IEEE, pp. 193–194. 2

[IGIA16] Issartel P., Guéniat F., Isenberg T., Ammi M.: Analysis of locally coupled 3d manipulation mappings based on mobile device motion. *CoRR abs/1603.07462* (2016). 12

[JH15] Jankowski J., Hachet M.: Advances in interaction with 3d environments. *Computer Graphics Forum 34*, 1 (2015), 152–190. 2

[KAD09] Kin K., Agrawala M., DeRose T.: Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics interface 2009* (2009), Canadian Information Processing Society, pp. 119–124. 7

[KCST05] Kruger R., Carpendale S., Scott S. D., Tang A.: Fluid integration of rotation and translation. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), ACM, pp. 601–610. 7

[KFM*01] Keefe D. F., Feliz D. A., Moscovich T., Laidlaw D. H., LaViola Jr J. J.: Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), ACM, pp. 85–93. 2

[KH10] Katzakis N., Hori M.: Mobile devices as multi-dof controllers. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on* (2010), IEEE, pp. 139–140. 10

[KH11] Knoedel S., Hachet M.: Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces* (Washington, DC, USA, 2011), 3DUI '11, IEEE Computer Society, pp. 75–78. 8

[KMB*11] Kin K., Miller T., Bollensdorff B., DeRose T., Hartmann B., Agrawala M.: Eden: a professional multitouch tool for constructing virtual organic environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 1343–1352. 8

[KP14] Kim T., Park J.: 3d object manipulation using virtual handles with a grabbing metaphor. *IEEE Computer Graphics and Applications*, 3 (2014), 30–38. 11, 25

[KTY97] Kiyokawa K., Takemura H., Yokoya N.: Manipulation aid for two-handed 3-d designing within a shared virtual environment. In *HCI (2)* (1997), pp. 937–940. 13

[LAFT12] Liu J., Au O. K.-C., Fu H., Tai C.-L.: Two-finger gestures for 6dof manipulation of 3d objects. *Computer Graphics Forum 31*, 7 (2012), 2047–2055. 8, 24

[LaV17] LaValle S. M.: *Virtual Reality*. Cambridge University Press, 2017. URL: http://vr.cs.uiuc.edu/. 2, 4, 17, 19

[LJKM*17] LaViola Jr J. J., Kruijff E., McMahan R. P., Bowman D. A., Poupyrev I.: *3D User Interfaces: Theory and Practice (2nd Edition)*. Addison-Wesley, 2017. 2

[MBJS97] Mine M. R., Brooks Jr F. P., Sequin C. H.: Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 19–26. 18

[MCG10a] Martinet A., Casiez G., Grisoni L.: The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on* (2010), pp. 115–118. doi:10.1109/3DUI.2010.5444709. 5, 7, 8, 16

[MCG10b] Martinet A., Casiez G., Grisoni L.: 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* (New York, NY, USA, 2010), VRST '10, ACM, pp. 111–118. doi:10.1145/1889863.1889888. 5, 7, 10, 16, 24

[MF11a] Mendes D., Ferreira A.: Evaluation of 3d object manipulation on multi-touch surfaces using unconstrained viewing angles. In *Human-Computer Interaction–INTERACT 2011*. Springer, 2011, pp. 523–526. 8

[MF11b] Moehring M., Froehlich B.: Effective manipulation of virtual objects within arm's reach. In *Virtual Reality Conference (VR), 2011 IEEE* (2011), IEEE, pp. 131–138. 2, 15

[MFA*14] Mendes D., Fonseca F., Araujo B., Ferreira A., Jorge J.: Mid-air interactions above stereoscopic interactive tables. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on* (2014), IEEE, pp. 3–10. 11, 15, 17, 25

[MJGJ11] Marquardt N., Jota R., Greenberg S., Jorge J. A.: The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III* (Berlin, Heidelberg, 2011), INTERACT'11, Springer-Verlag, pp. 461–476. 11

[MLF11] Mendes D., Lopes P., Ferreira A.: Hands-on interactive tabletop lego application. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (New York, NY, USA, 2011), ACE '11, ACM, pp. 19:1–19:8. doi:10.1145/2071423.2071447. 2, 8, 16, 24

[MM95] Mapes D. P., Moshell J. M.: A two-handed interface for object manipulation in virtual environments. *Presence: Teleoperators & Virtual Environments 4*, 4 (1995), 403–416. 11

[MRFJ16] Mendes D., Relvas F., Ferreira A., Jorge J.: The benefits of dof separation in mid-air 3d object manipulation. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (2016), ACM, pp. 261–268. 15, 17, 25

[MSH04] Mujber T. S., Szecsi T., Hashmi M. S.: Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology 155* (2004), 1834–1838. 2

[MYC14] Mine M., Yoganandan A., Coffey D.: Making vr work: building a real-world immersive modeling application in the virtual world. In *Proceedings of the 2nd ACM symposium on Spatial user interaction* (2014), ACM, pp. 80–89. 12

[MYC15] Mine M., Yoganandan A., Coffey D.: Principles, interactions and devices for real-world immersive modeling. *Computers & Graphics 48* (2015), 84–98. 2, 12

[MZB12] Möllers M., Zimmer P., Borchers J.: Direct manipulation and the third dimension: co-planar dragging on 3d displays. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (New York, NY, USA, 2012), ITS '12, ACM, pp. 11–20. doi:10.1145/2396636.2396639. 9

[NBBW09] Nacenta M. A., Baudisch P., Benko H., Wilson A.: Separability of spatial manipulations in multi-touch interfaces. In *Proceedings of Graphics Interface 2009* (Toronto, Ont., Canada, Canada, 2009), GI '09, Canadian Information Processing Society, pp. 175–182. 7

[ND13] Nguyen T. T. H., Duval T.: Poster: 3-point++: A new technique for 3d manipulation of virtual objects. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on* (2013), IEEE, pp. 165–166. 5, 14, 17

[NDP14] Nguyen T. T. H., Duval T., Pontonnier C.: 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* (2014), p. 8. 5, 14, 17, 25

[NOJ87] Nielson G. M., Olsen Jr D. R.: Direct manipulation techniques for 3d objects using 2d locator devices. In *Proceedings of the 1986 workshop on Interactive 3D graphics* (1987), ACM, pp. 175–182. 3, 5, 23

[ocu17] Oculus Rift. www.oculus.com/rift, 2017. [Online; accessed 3-January-2018]. 10

[Osa08] Osawa N.: Two-handed and one-handed techniques for precise and efficient manipulation in immersive virtual environments. In *Advances in Visual Computing*, Bebis G., Boyle R., Parvin B., Koracin D., Remagnino P., Porikli F., Peters J., Klosowski J., Arns L., Chun Y. K., Rhyne T.-M., Monroe L., (Eds.). Springer, Berlin, Heidelberg, 2008, pp. 987–997. 13, 17, 25

[PBWI96] Poupyrev I., Billinghurst M., Weghorst S., Ichikawa T.: 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* (1996), ACM, pp. 79–80. 12, 25

[psm17] PlayStation Move. www.playstation.com/en-us/explore/accessories/vr-accessories/playstation-move, 2017. [Online; accessed 3-January-2018]. 10

[PSP99] Pierce J. S., Stearns B. C., Pausch R.: Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics* (1999), ACM, pp. 141–145. 13, 25

[PSR*15] Perelman G., Serrano M., Raynal M., Picard C., Derras M., Dubois E.: The roly-poly mouse: Designing a rolling input device unifying 2d and 3d interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 327–336. 6

[RDH09] Reisman J. L., Davidson P. L., Han J. Y.: A screen-space formulation for 2d and 3d direct manipulation. In *Proceedings of the*

22nd annual ACM symposium on User interface software and technology (New York, NY, USA, 2009), UIST '09, ACM, pp. 69–78. doi:10.1145/1622176.1622190. 7, 8, 9, 16, 24

[RH92] ROBINETT W., HOLLOWAY R.: Implementation of flying, scaling and grabbing in virtual worlds. In *Proceedings of the 1992 symposium on Interactive 3D graphics* (1992), ACM, pp. 189–192. 10

[SBAG16] SIMEONE A. L., BULLING A., ALEXANDER J., GELLERSEN H.: Three-point interaction: combining bi-manual direct touch with gaze. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (2016), ACM, pp. 168–175. 3

[SCP95] STOAKLEY R., CONWAY M. J., PAUSCH R.: Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 265–272. 13, 25

[SDGK16] SPEICHER M., DAIBER F., GEHRING S., KRÜGER A.: Exploring 3d manipulation on large stereoscopic displays. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays* (2016), ACM, pp. 59–66. 12

[SGH*12] SONG P., GOH W. B., HUTAMA W., FU C.-W., LIU X.: A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 1297–1306. doi:10.1145/2208516.2208585. 3, 11, 12, 14, 15, 17, 25

[Sho92] SHOEMAKE K.: Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface* (1992), vol. 92, pp. 151–156. 5, 6, 8, 9, 23

[Sim16] SIMEONE A. L.: Indirect touch manipulation for interaction with stereoscopic displays. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on* (2016), IEEE, pp. 13–22. 10, 24

[SJT*12] SCHULTHEIS U., JERALD J., TOLEDO F., YOGANANDAN A., MLYNIEC P.: Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3d tasks. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on* (2012), IEEE, pp. 117–124. 14

[SSB08] SCHMIDT R., SINGH K., BALAKRISHNAN R.: Sketching and composing widgets for 3d manipulation. *Computer Graphics Forum 27*, 2 (2008), 301–310. 3, 8

[SVH11] STROTHOFF S., VALKOV D., HINRICHS K.: Triangle cursor: interactions with objects above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2011), ITS '11, ACM, pp. 111–119. doi:10.1145/2076354.2076377. 9, 10, 24

[VCB09] VEIT M., CAPOBIANCO A., BECHMANN D.: 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* (2009), ACM, pp. 51–58. 14

[VDLS02] VAN DAM A., LAIDLAW D. H., SIMPSON R. M.: Experiments in immersive virtual reality for scientific visualization. *Computers & Graphics 26*, 4 (2002), 535–555. 2

[VSC15] VUIBERT V., STUERZLINGER W., COOPERSTOCK J. R.: Evaluation of docking task performance using mid-air interaction techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (2015), ACM, pp. 44–52. 14

[WB08] WILKES C., BOWMAN D. A.: Advantages of velocity-based scaling for distant 3d manipulation. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology* (2008), ACM, pp. 23–29. 13, 25

[WCOM15] WU S., CHELLALI A., OTMANE S., MOREAU G.: Touchsketch: a touch-based interface for 3d object manipulation and editing. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology* (2015), ACM, pp. 59–68. 9, 17, 24

[WIH*08] WILSON A. D., IZADI S., HILLIGES O., GARCIA-MENDOZA A., KIRK D.: Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), ACM, pp. 67–76. 4

[WMSB98] WANG Y., MACKENZIE C. L., SUMMERS V. A., BOOTH K. S.: The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1998), ACM Press/Addison-Wesley Publishing Co., pp. 312–319. 7

[WP09] WANG R. Y., POPOVIĆ J.: Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 63:1–63:8. doi:10.1145/1576246.1531369. 11

[WPP11] WANG R., PARIS S., POPOVIĆ J.: 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2011), UIST '11, ACM, pp. 549–558. doi:10.1145/2047196.2047269. 11, 17

[WW11] WIGDOR D., WIXON D.: *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011. 7

[ZFS97] ZELEZNIK R. C., FORSBERG A. S., STRAUSS P. S.: Two pointer input for 3d interaction. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (1997), ACM, pp. 115–120. 3, 5, 23

Technique	Environment Properties		Transformations												
	Display Mapping	Tracking Space	Separation	Translation				Rotation				Scaling			
				Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Triad cursor [NOJ87]	SC	2D Separated	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	3	1	Remapped	1	3	1
Two Pointer [ZFS97]	SC	2D Separated	Partial: {T},{T,R},{R},{T,S}	Exact	1-2	3	2	Exact	2	3	1	Distance	2	3	1
Handle box [Hou92]	SC	2D Separated	Total: {T},{R}	Hybrid	1	3	1	Remapped	1	1	1	No control			
Virtual Handles [CSH*92]	SC	2D Separated	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	3	1	Remapped	1	3	1
Arcball [Sho92]	SC	2D Separated	Only {R}	No control				Remapped	1	3	1	No control			

Table 1: Classification of techniques for manipulating 3D virtual objects with desktop interfaces (SC: screen constrained, SW: stereoscopic window, CP: number of contact points required, TD: total transformation DOFs supported, MD: minimum explicitly simultaneously controlled DOFs).

Technique	Environment Properties		Transformations												
	Display Mapping	Tracking Space	Separation	Translation				Rotation				Scaling			
				Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Sticky Fingers [HtCC09]	SC	2D Co-located	Partial: {Txy},{T,Rz},{T,R}	Exact	1-3	3	2	Exact	2-3	3	1	No control			
Screen-space [RDH09]	SC	2D Co-located	Partial: {Txy},{T,Rz},{T,R}	Exact	1+	3	2	Exact	2+	3	1	No control			
DS3 [MCG10b]	SC	2D Co-located	Total: {T},{R}	Hybrid	1-2	3	2	Hybrid	2+	3	1	No control			
Liu et al. [LAFT12]	SC	2D Co-located	Partial: {T,Rz},{Rxy}	Exact	2	3	2	Exact	2	3	1	No control			
tBox [CDH11]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	1	3	1	Remapped	1	1	1	Remapped	1	3	1
LTouchIt [MLF11]	SC	2D Co-located	Total: {T},{R}	Exact	1	3	2	Remapped	2	3	1	No control			
Au et al. [ATF12]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	2	3	1	Remapped	2	3	1	Distance	2	1	1
GimbalBox [BHA12]	SC	2D Co-located	Total: {T},{R}	Remapped	1	3	2	Remapped	1-2	1	1	No control			
TouchSketch [WCOM15]	SC	2D Co-located	Total: {T},{R},{S}	Remapped	2	3	1	Remapped	2	3	1	Distance	2	3	1
Triangle Cursor [SVH11]	SW	2D Co-located	None: {T,R}	Hybrid	2	3	3	Exact	2	1	1	No control			
Toucheo [HBCdlR11]	SW	2D Separated	Partial: {T},{Txy,Rz,S},{Rxy},{S}	Hybrid	1-2	3	1	Hybrid	1-2	3	1	Hybrid	1-2	3	1
Indirect6 [Sim16]	SW	2D Separated	Total: {T},{R}	Hybrid	1-2	3	2	Remapped	2	3	1	No control			
Void Shadows [GVH14]	SW	2D Co-located	Partial: {Txy},{T,Rz}	Exact	1-2	3	2	Exact	2	1	1	No control			

Table 2: Classification of techniques for manipulating 3D virtual objects with touch-based interfaces (SC: screen constrained, SW: stereoscopic window, CP: number of contact points required, TD: total transformation DOFs supported, MD: minimum explicitly simultaneously controlled DOFs).

Technique	Environment Properties			Transformations												
	Display	Tracking	Hands/DOF	Separation	Translation				Rotation				Scaling			
	Type	Space	Tracked		Mapping	CP	TD	MD	Mapping	CP	TD	MD	Mapping	CP	TD	MD
Simple Virtual Hand [BKLJP04]	RR	3D Co-located	1/6	None: {T,R}	Exact	1	3	3	Exact	1	3	3	No control			
In the Air [HIW*09]	SC	3D Separated	1/4	None: {T,R}	Exact	1	3	3	Exact	1	1	1	No control			
Air-TRS [ACJH13]	SW	3D Co-located	2/3	Partial: {T},{T,R,S}	Exact	1-2	3	3	Exact	2	3	3	Distance	2	1	1
VHGM [KP14]	RR	3D Co-located	1/6	None: {T,R}	Exact	1	3	3	Exact	1	3	3	No control			
Handle Bar [SGH*12]	SC	3D Separated	2/3	Partial: {T,Ryz},{Rx}	Exact	2	3	3	Hybrid	2	3	1	Distance	2	1	1
Spindle+Wheel [CW15]	SW	3D Separated	2/6	None: {T,R}	Exact	2	3	3	Hybrid	2	3	3	Distance	2	1	1
Crank Handle [BMA*14]	SW	3D Separated	1/3	Total: {T},{R}	Exact	1	3	3	Remapped	1	3	1	No control			
Grasping Object [BMA*14]	SW	3D Separated	1/3	Partial: {T},{T,R}	Exact	1	3	3	Remapped	1	3	3	No control			
3-DOF Hand [MFA*14]	SW	3D Co-located	2/6	Partial: {T},{T,R,S}	Exact	1	3	3	Exact	1	3	3	Distance	2	1	1
6-DOF Hand [MFA*14]	SW	3D Co-located	2/6	None: {T,R},{T,R,S}	Exact	1	3	3	Exact	1	3	3	Distance	2	1	1
PRISM [FKK07]	RR	3D Co-located	1/6	None: {T,R}	Scaled N:1	1	3	3	Scaled N:1	1	3	3	No control			
Viewpoint Adjustment [Osa08]	RR	3D Co-located	2/6	None: {T,R}	Exact	1	3	3	No control				No control			
7 Handle [NDP14]	RR	3D Co-located	2/3	None: {T,R}	Remapped	1-2	3	3	Remapped	1-2	3	1	No control			
Widgets [MRFJ16]	RR	3D Co-located	1/3	Total: {T},{R}	Remapped	1	3	1	Remapped	1	3	1	No control			
Go-Go [PBWI96]	RR	3D Co-located	1/6	None: {T,R}	Scaled 1:N	1	3	3	Exact	1	3	3	No control			
HOMER [BH97, WB08]	RR	3D Co-located	1/6	None: {T,R}	Scaled 1:N	1	3	3	Exact	1	3	3	No control			
Worlds in Miniature [SCP95]	RR	3D Co-located	2/6	None: {T,R}	Remapped	1	3	3	Remapped	1	3	3	No control			
Voodoo Dolls [PSP99]	RR	3D Co-located	2/6	None: {T,R}	Remapped	1	3	3	Remapped	1	3	3	No control			

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