Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Reaching a New Dimension: Using Tangibles for 3D Object Rotation

Master's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
David Asselborn

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 03.09.2018
Submission date: 28.09.2018

Eidesstattliche Versicherung

____________________________ ____________________________

Name, Vorname Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum Unterschrift

Contents

List of Figures

List of Tables

Abstract

Tangibles on tabletops increase speed, accuracy, and awareness of each other's actions. Providing haptic feedback they can bridge the gap between physical object manipulation and its virtual counterpart. But until now they have been mostly used on-surface to provide input. With this work, we are going to open a new interaction space by additionally bringing tangibles off-surface. As a first use case, we propose to use them for 3D object manipulation controlling translation on-surface and rotation off-surface in midair.

For this purpose tangible hardware and software are extended. We describe how the new communication protocol allows for a modular data exchange and how 3D content is embedded into the existing 2D tangible framework. Designing the transition between on-surface and off-surface interaction is crucial, therefore we develop two variants for comparison. One is continuously mirroring the tangible orientation to the controlled virtual object, the other one is allowing to temporally fix the virtual object orientation with a toggle. In addition to that, we analyze if a perceptual analogy between physical object input and virtual object output could positively influence user performance. For comparison, we also implement two other input methods inspired by related work. First, touch input (2DoF) is supported using a rotation gizmo and the virtual trackball technique. Second, 3D mouse input (6DoF) can be used as designated hardware.

Our conducted user study reveals that even if the 3D mouse performed fastest, both tangible methods were more intuitive to use for many users. On average, they led to less translations and more rotations of the controlled object than touch or 3D mouse. Additionally, a perceptual analogy between input and output had a positive impact on user performance in tangible mode. Monitoring the user progress in detail, we furthermore could observe different tactics highlighting properties of each mode. Consequently, these insights allow us to derive design recommendations. By reaching a new dimension of tangible interaction, tangibles could be used for a variety of new tasks in future.

Überblick

Tangibles auf Tischbildschirmen erhöhen die Geschwindigkeit, Genauigkeit, und Aufmerksamkeit für gegenseitige Aktionen. Durch haptisches Feedback schließen sie die Lücke zwischen physischer Objektmanipulation und ihrer virtuellen Entsprechung. Bisher wurden sie vorwiegend direkt auf dem Bildschirm für Eingaben genutzt. Mit dieser Arbeit öffnen wir einen neuen Interaktionsraum, indem wir Tangibles zusätzlich über dem Bildschirm nutzen. Als ersten Anwendungsfall möchten wir diese für 3D-Objektmanipulation einsetzen, die Translation kann auf dem Bildschirm, die Rotation in der Luft gesteuert werden.

Zu diesem Zweck werden Tangible Hardware und Software erweitert. Wir beschreiben, wie das neue Kommunikationsprotokoll einen modularen Datenaustausch ermöglicht und wie 3D-Inhalte in das bestehende 2D-Tangible-Framework eingebettet werden. Die Gestaltung des Übergangs zwischen Interaktion auf und über dem Bildschirm ist entscheidend, daher entwickeln wir zwei Varianten zum Vergleich. Die eine spiegelt die Orientierung kontinuierlich vom Tangible auf das virtuelle Objekt, die andere erlaubt die Orientierung des virtuellen Objekts temporär zu fixieren. Darüber hinaus analysieren wir, ob eine Wahrnehmungsanalogie zwischen physischer Eingabe und virtueller Ausgabe die Nutzerleistung positiv beeinflussen könnte. Zum Vergleich implementieren wir außerdem zwei weitere Eingabemethoden, angelehnt an existierende Lösungen. Erstens wird Touch-Eingabe (2 Freiheitsgrade) mit Hilfe eines Rotations-Widgets unterstützt. Zweitens kann eine 3D-Maus (6 Freiheitsgrade) als speziell für diesen Zweck existierende Hardware genutzt werden.

Unsere Anwenderstudie zeigt, dass, selbst wenn die 3D-Maus am schnellsten arbeitete, beide Tangible Methoden für viele Anwender intuitiver zu bedienen waren. Im Durchschnitt führten sie zu weniger Translationen und mehr Rotationen des kontrollierten Objekts als Touch oder 3D-Maus. Darüber hinaus waren Nutzer schneller wenn sie ein dem Tangible nachempfundenes Objekt mit selbigem manipulieren konnten. Bei der detaillierten Analyse des Benutzerfortschritts konnten wir außerdem verschiedenen Taktiken beobachten, welche die Eigenschaften der einzelnen Modi hervorheben. Aus diesen Erkenntnissen können dann Gestaltungsempfehlungen abgeleitet werden. Durch das Erreichen einer neuen Dimension der Tangible Interaktion könnten diese in Zukunft für eine Vielzahl neuer Aufgaben genutzt werden.

Acknowledgements

My first thanks go to my supervisor Christian Cherek for the pleasant collaboration throughout the past months. Additionally, I want to especially thank Marco Ruchay for helping with the tangible construction.

Secondly, thank you to all participants of the user study for your time and valuable input.

Thirdly, I want to thank Prof. Dr. Jan Borchers and Prof. Dr. Ulrik Schroeder for examining the thesis.

Last but not least, thanks to my family and friends for supporting me.

Conventions

Throughout this thesis we use the following conventions.

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition: *Excursus*

Source code and implementation symbols are written in typewriter-style text.

```
myClass
```

The whole thesis is written in American English.

Chapter 1

Introduction

A tangible user interface (TUI) allows people to interact with virtual content through physical objects, so-called tangibles. Tangibles can be customized both in their outer shape and in their embedded hardware. This enables them to act as representatives for all kinds of virtual entities. Placed on a multi-touch tabletop, tangibles create touch patterns which can be detected to provide input to the digital content underneath. On top of that, tangibles give distinct haptic feedback which would otherwise be missing when interacting with a flat surface only. Consequently, tangibles can bridge the gap between real world object manipulation and its virtual counterpart.

Bridge the gap between physical and virtual objects by using tangibles

Up to this point, tangibles have been mostly used on-screen in a two-dimensional context (e.g. by Tuddenham et al. [2010]). Moving around virtual sprites on a screen with rotation limited to their z-axis, tangibles stay on the tabletop to provide input. Designed for a specific purpose and used in a space multiplex scenario, they can be superior to general input devices in a time multiplex scenario (Fitzmaurice and Buxton [1997]). It has already been shown that tangibles designed as rotary knobs allow for a more precise rotation control compared to all virtual input (Voelker et al. [2015b]). In addition to that, tangibles can increase awareness of each other's actions when working together collaboratively on larger tabletops (Cherek et al. [2018]).

Tangibles are currently used on-surface in a two-dimensional context

Use tangibles off-surface in a three-dimensional context

On the basis of these promising results, a whole new dimension will be added to the interaction. By equipping the tangible with an embedded tracking device its orientation can be detected. This introduces a new way of providing input, still integrated into the already established framework of tangibles. As a result, they can be used both on- and off-surface opening a wide range of new possibilities. Putting this idea into practice, the new type of tangibles is going to be used for object rotation tasks in 3D in midair combined with object translation tasks in 2D on the tabletop.

Definition: *Degrees of freedom (DoF)*

DEGREES OF FREEDOM (DOF):
In general, this term describes the number of independent system parameters. In the context of this work, it is used to define distinct input dimensions. For example a conventional computer mouse has 2DoF: its movement in the x- and in the y-dimension.

Challenge of 3D object manipulation

With the improved graphics power over time, three-dimensional scenes are now used on both stationary and mobile devices for all kinds of purposes like education, infotainment and gaming. But usually only general input methods are available, for example conventional mouse and keyboard input originally developed for 2D window management. When manipulating virtual objects, dimensions of input have to be mapped to dimensions of output in some way. A conventional computer mouse or single touch only provide 2DoF. This makes designing the user interaction in three dimensions a challenge as a direct mapping of input to output dimensions is not possible (Jacob and Sibert [1992]). Especially novice users could then notice the lack of straight forward interaction with 3D objects.

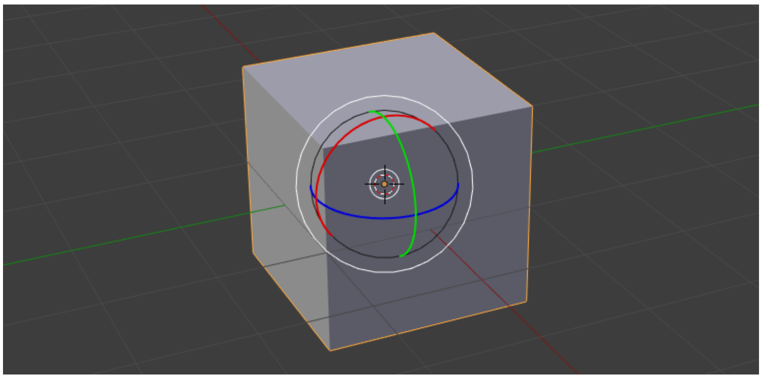


Figure 1.1: Rotation gizmo in Blender for separate axis control working with 2DoF computer mouse input (Blender Foundation).

3D object rotation with general-purpose hardware requires conversion in software

Some applications therefore limit their adjustable rotation axes, only allowing to manipulate two of them. Of course, this is only applicable in some application domains. For full-featured 3D graphics tools like Blender, Maya, Cinema 4D, or Shapr3D (iOS) however, manipulation of all axes is a requirement. Therefore, they usually split up the rotation into multiple steps instead. First, users have to select the axis they want to manipulate on a so-called rotation gizmo as shown in figure 1.1. Subsequently, only changes on that one rotation axis are allowed, making it a time multiplex process. In addition to that, the virtual trackball technique (Henriksen et al. [2004]) is broadly used, simulating a direct grab on the object. It puts a virtual sphere on the object onto which any point in 2D can be mapped. Dragging on this sphere can then be translated to rotation around multiple axes. As this mode manipulates all axes at once with dependencies on each other, it is however only helpful in some situations. On touch based systems multi-touch gestures can be used to increase the variety of input schemes, but they have to be explicitly learned by the user.

3D object rotation with designated hardware allows for direct mapping from input to output

Because of that, conventional established input methods are not well suitable for flexible three-dimensional rotation adjustments. As already mentioned, Fitzmaurice and Buxton [1997] favor separate input devices in a space multiplex scenario. This contradicts the universal usage of a standard computer mouse just splitting up the rotation tasks into multiple steps. Instead, designated input hardware as the 3D mouse can be used to allow for a direct mapping of 6DoF input to 3D object rotation and even translation. With this type of input device, no different modes or simulated grabbing is needed. However, the 3D mouse uses a generic knob to control all axes in a relative way.

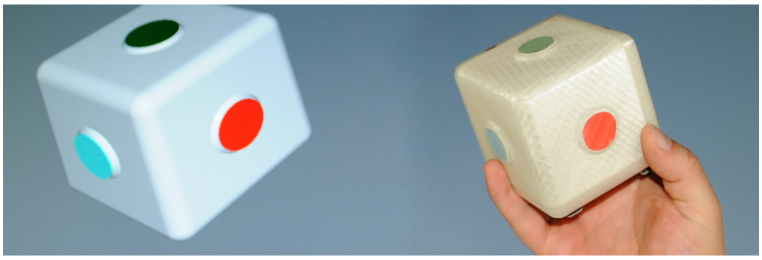


Figure 1.2: We propose to use tangibles off-surface mirroring their orientation to a virtual counterpart.

Mirror orientation of tangible to virtual object

Especially in immerse environments tangibles are already considered an option for 3D object manipulation (Cannavò et al. [2017], Rodrigues et al. [2017], Zielinski et al. [2017]). With tangibles a direct mapping of 3DoF input to all three rotation axes is possible. While the 3D mouse only allows for a relative input control, the tangible orientation can be used as an absolute control for the virtual counterpart. What has been used in a two-dimensional context on the tabletop is now extended to work in a three-dimensional context in midair.

Combine translation on-surface and rotation off-surface

In contrast to already proposed solutions, we will use tangibles both on- and off-surface examining how to design for this new way of interaction. Several options to perform object rotation and translation with tangibles are possible. One tangible could be used on- and off-surface or two tangibles could be used in parallel. With our work we will focus on using one tangible for both tasks to emphasize the relation between physical and virtual object and focus on the transition between on- and off-surface interaction. Here, the tangible properties in combination with the surface work as a physical constraint, because it is only possible to position the object on the tabletop in certain orientations. A cube for example could be placed solely on one of its six faces. If this forced orientation change is transmitted to the virtual object as well, it could cause unwanted rotations. Because of that, we will also implement an optional toggle button to allow locking the orientation temporally in its current state and examine its impact on user performance.

Definition: *Mapping*

MAPPING:
A mapping describes the relationship between input controls and output results. Natural mappings are understood immediately due to their use of e.g. spatial or perceptual analogies.

- Spatial analogy: input controls are arranged the same way as output counterparts.
- Perceptual analogy: input controls are an imitation of the controlled object.

Every perceptual analogy is also a spatial analogy but not vice versa.

Spatial versus perceptual analogy

Another interesting aspect to analyze will be the impact of different analogy levels between tangible and virtual object. Transferring the rotation of each tangible axis to the same axis of a virtual object leads to a spatial analogy between input and output. If the tangible is even used to control a virtual representation of itself, a perceptual analogy between input and output can be achieved. In theory, sensor values of the embedded tangible hardware can be used for an absolute or relative control. If the tangible imitates the virtual object it controls, the tangible orientation should always be mirrored in an absolute manner. Otherwise the one-to-one mapping between tangible and virtual object would not be valid anymore.

Outline

Our work is going to open a new interaction space by bringing tangibles off-surface. First, it is taken a look at related work concerning 3D object manipulation, tangibles and the existing software framework used for them. Next, we proceed to a detailed description on how the tangible software and hardware is extended to work for the new task of manipulating 3D objects. To later contrast our solution to other input methods, the implementation of touch and 3D mouse support on the tabletop is described as well. All these input methods are compared to each other in a user study, whose setup and procedure is presented. Subsequently, we discuss study results and derive design recommendations for working with the new type of tangibles. A summary and suggestions for future work round this thesis off.

Chapter 2

Related Work

Overview

First, it is summarized how the task of 3D object manipulation is solved by others. Focusing on rotation, its origins, developments and different approaches for various media types are presented. Second, it is proceeded to tangible research. What are tangibles good for? In which contexts are they currently used? By that, their usefulness for the described rotation task is derived. Third, it is then taken a detailed look into the existing framework and its architecture used for tangibles to understand where to hook in with the implementation. What can the framework already contribute? Where does it have to be extended? All this should provide the basis for the main part of this thesis.

2.1 3D Object Manipulation

The one perfect solution to manipulate 3D objects does not yet exist

To manipulate a virtual object in all three dimensions first may sound like a simple task, but the wide range of research activities in this area shows, how different it can be approached. Various virtual and tangible user interfaces are proposed. Starting with virtual and physical trackballs, continuing with extensions of the conventional computer mouse and finishing with the current challenge of mobile and immerse virtual environments, it is shown that the one perfect solution to solve this task does not yet exist.

2.1.1 Virtual and Physical Trackball

Virtual trackball translates between 2DoF input and 3D rotation

Early approaches to rotate 3D objects rely on already existing input hardware. A commonly used one described by Chen et al. [1988] is to put a virtual trackball around the object, translating between 2DoF input of a conventional computer mouse and object rotation around three axes. Shoemake [1992] proposes another adaption of the virtual trackball using quaternions. A comparison between three approaches including the two above is performed by Henriksen et al. [2004], who focus on the mathematical characteristics of the different solutions. More on that in chapter 3.3, as such a solution is also implemented for our user study.

Physical trackball leads to shorter task completion times without sacrificing accuracy

However, Hinckley et al. [1997] show that using a physical trackball can have a positive effect on task completion time of 3D rotation tasks. Compared to using a virtual trackball, designated hardware allows for a direct mapping of input dimensions to output dimensions. They construct two variants of an absolute rotation controller: a ball and a tracker. Both work in a similar way like our tangible, mirroring their orientation to the virtual object. In a conducted user study, participants should use the ball, the tracker or two versions of the virtual trackball to match the orientation of a controlled object with a goal object. To also measure accuracy, participants should indicate when they feel like having completed the task. As a result, users were significantly faster using the physical devices, without sacrificing accuracy.

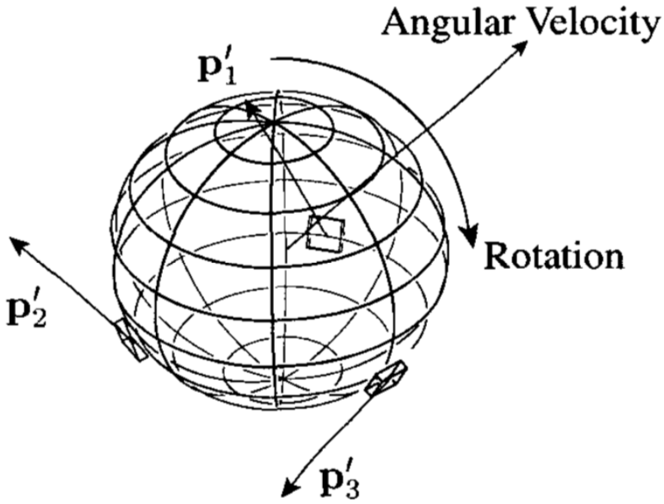


Figure 2.1: A 3D trackball using three optical sensors (Kim et al. [2001]).

There exist several ways to track the rotation of a physical trackball. The mentioned 3D ball uses a magnetic tracker, Kim et al. [2001] however analyze how many optical sensors are required to construct a physical trackball offering 3DoF. The version using three sensors arranged as shown in figure 2.1 represents a good trade off between cost and accuracy. It sets their input device apart from other commercially available trackballs at that time, which could only provide 2DoF, while still being relatively cheap to construct. Compared to a Magellan/SpaceMouse in a user study matching 3D object orientations their solution yielded to 30~40% faster task completion times.

Construction of physical trackball using optical sensors

Likewise focusing on how to track the rotation in all three dimensions with sensors, a more recent publication by Lin et al. [2015] proposes another way of constructing a 3D trackball. Using RFID tags on predefined positions attached to the ball, its translation and rotation in all three dimensions can be derived by measuring their relative signal strength. The RFID tags are inexpensive and work without batteries.

Construction of physical trackball using RFID tags

2.1.2 3D Mouse

(a) (b)

Figure 2.2: The SpaceMouse in a compact (a) and regular (b) format as presented by 3Dconnexion[1].

Stationary (3D mouse) versus non-stationary (3D ball) devices

A non-stationary trackball as proposed in the previous chapter 2.1.1 has the disadvantage of being operated in the air without a surface on which the device could rest on. Therefore, it could be argued that it requires more attention starting and stopping to use a non-stationary device versus a stationary device which is operated on a surface. For example, the conventional computer mouse is placed on a surface all the time, so it can be grabbed and released easily. Influenced by that, manifold variations of stationary 3D input devices are proposed, which are more closely coupled to the traditional computer mouse.

SpaceMouse provides 6DoF attached to one knob

The SpaceMouse by 3Dconnexion is a commercially available product. It allows for 6DoF attached to a knob which can be pushed, pulled, panned parallel to the surface, tilted and torqued. Several configurations are available, two examples can be seen in figure 2.2. The concept of the SpaceMouse motivated adaptations based on gestures (Kurpjuhn et al. [1999]) and has been proven to also help people with disabilities navigating in 3D compared to using standard mouse and keyboard input (Martins et al. [2015]).

[1]`www.3dconnexion.com/products/spacemouse.html`

Another way of adding an additional input dimension to the mouse is by using two trackballs instead of one, as presented by MacKenzie et al. [1997]. The design of their mouse can stay much closer to the already known conventional one, which could positively influence user acceptance. Using the two trackballs, it can use their relative movement in relation to each other to derive how the mouse body is oriented. This can be used as an additional input parameter besides the two-dimensional relative movement of the whole body.

Computer mouse can provide 3DoF by using two trackballs

Figure 2.3: 3D mouse using two trackballs designed by Fallman et al. [2007].

Designing a 3D mouse provides multiple challenges to solve. Not only technical aspects have to be considered, the 3D mouse also has to be designed in a way to be easily usable over a longer time period. Based on the two trackball approach, Fallman et al. [2007] suggest a mouse design as presented in figure 2.3. It picks up design elements of the familiar 2D mouse, while also emphasizing its two contact points to the surface.

Design of 3D mouse should emphasize its capabilities

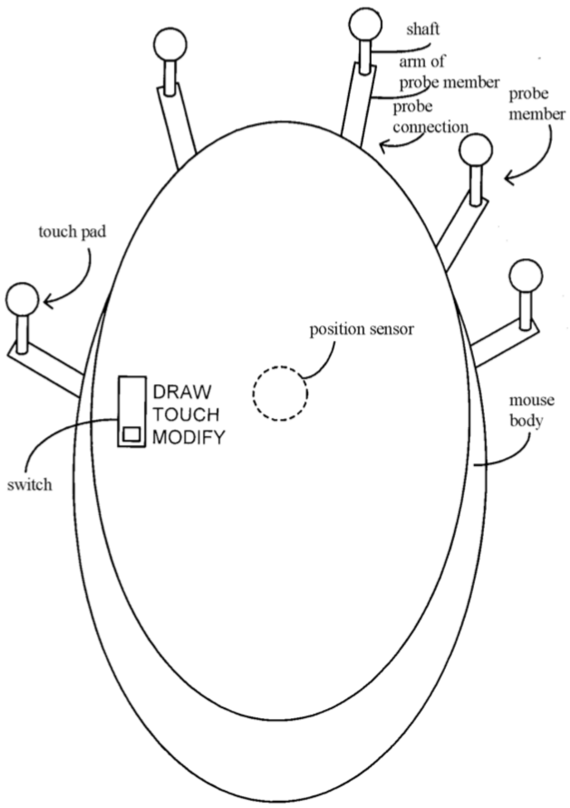


Figure 2.4: 3D mouse to feel and modify 3D objects as proposed by Chen and Brown [2005].

Extend computer mouse with probe members to feel and modify 3D shape

Another hardware proposal to feel and manipulate virtual 3D content while minimizing the requirement of mathematical understanding is presented by Chen and Brown [2005]. They describe the idea of a 3D mouse different from today's commonly used SpaceMouse. Instead of having a knob providing 6DoF, their device reshapes the virtual content in a physical form. So-called probe members which are shown in figure 2.4 should allow the user to feel the shape of three-dimensional surfaces. One for each finger they can change height depending on the virtual content and also allow for input by the user. In different modes the user should be able to draw new content, touch existing content or modify it. This mechanism is attached to a traditional mouse body which can still be moved in the same way as its 2D counterpart.

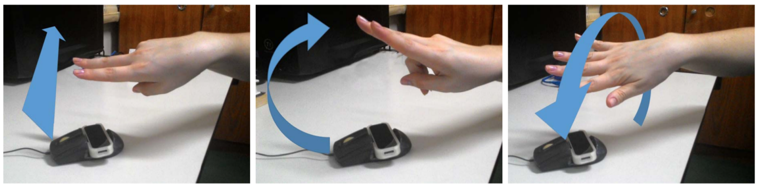


Figure 2.5: 2DoF mouse input extended with gestures (Franz et al. [2016]).

Extend computer mouse with gesture recognition

Combining traditional mouse input with extended capabilities is also proposed by Franz et al. [2016]. They suggest using additional 3D gestures when using the mouse as shown in figure 2.5. This could be used to switch between applications, change the volume or perform other typical secondary tasks without having to leave focus of the main task. They do not primarily consider using the 3D gestures for object rotation, but the gesture recognition could also be extended to serve that purpose.

2.1.3 Voice Control

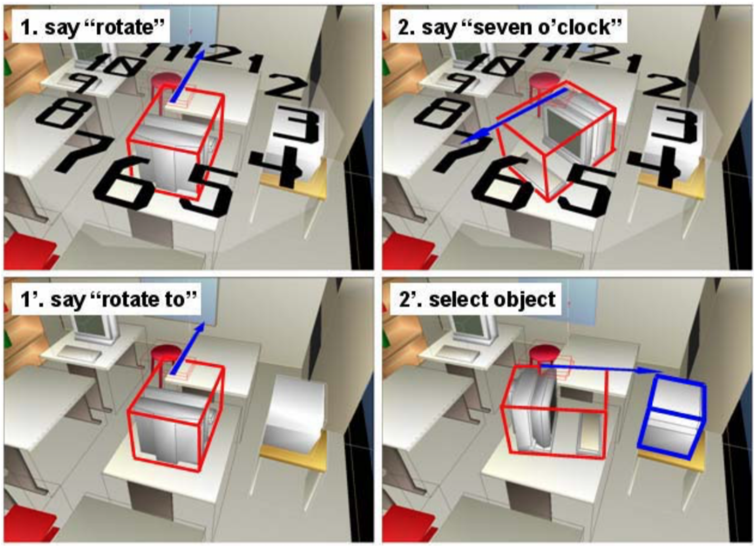


Figure 2.6: Using voice commands to rotate object based on clock analogy (top) or with respect to other object (bottom) (Fukutake et al. [2005]).

Use voice recognition to control object rotation

Following a completely different path, Fukutake et al. [2005] consider using voice commands for layout tasks including object rotation. Building up on their automatic 3D layouting tool, they introduce different commands for object control. Based on the analogy between rotation around one axis and the layout of a clock, it can therefore for example be commanded to rotate to seven o'clock. As transferring a large amount of precise information via voice is inconvenient and time consuming, they also introduce the possibility to change the orientation of an object in relation to other objects. Both modes of rotation can be seen in figure 2.6.

2.1.4 Tracking in Immerse Environments

Challenge of object manipulation in immerse environments

With the rise of Augmented Reality (AR) and Virtual Reality (VR), immersive 3D modeling introduces new possibilities and restrictions. Displaying content on large wall-sized screens, in AR or in VR is combined with room-scale tracking. By that, hands or specific controllers can be tracked to provide 6DoF.

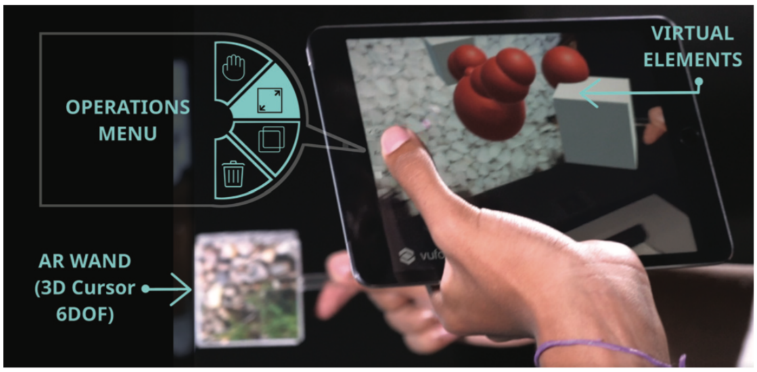
Track hand gestures for object manipulation

For example, Kim et al. [2005] suggest to use hand tracking in combination with a set of hand gestures. With that approach, they want to make the interaction with virtual objects as direct as possible. To start rotating an object, they suggest to open the hand and pinch. The rotation is then controlled in all three dimensions by the subsequent hand rotation. In a conducted user study, participants complained about a lack of control especially for precise adjustments. To tackle that problem, the authors plan to combine hand tracking with other modes of interaction, like virtual gizmos for precise control and textual menus to indicate and change modes.

Specify rotation using three points: two defining the rotation axis, the other one defining the rotation around it

Nguyen and Duval [2013] introduce a metaphor for manipulating objects in 6DoF by modifying three adjustment points attached to the object. With their metaphor, they want to tackle the challenge of performing precise adjustments and lower the burden of keeping hands motionless to fix a certain orientation. The relation between the three points, including their so called barycenter, is used to derive the desired object control. For rotation, the line between two points is used as the rotation axis while the third point then defines the rotation around it. In a preliminary study their 3-Point++ technique showed no significant difference compared to a direct 6DoF method. They assume that their prototype is still too complicated to use.

(a) Tangible rotary knob on the left to adjust one-dimensional parameter based on selection in on-screen menu on the right (Cannavò et al. [2017]).



(b) Tangible wand to provide 6DoF in different modes based on selection in on-screen operations menu (Rodrigues et al. [2017]).

Figure 2.7: Two different tangible user interfaces in AR using a rotary knob (a) or a wand (b).

Rotary knob in AR allowing for fine adjustments in one dimension

3D Object manipulation in AR on mobile devices introduces an additional challenge as Cannavò et al. [2017] state. On the compact screen of a mobile device control elements to switch between modes and the content itself have to be displayed, competing for screen space. They combine virtual parameter and object selection on the screen with a rotary tangible which provides haptic feedback and can be used for finer adjustments. The type of manipulation for the one-dimensional tangible input has to be explicitly selected to distinguish between the mode of operation. The interface can be seen in figure 2.7 (a). It puts knowledge in the world, but it could be criticized that it also competes with the virtual content itself. On a small mobile screen one could argue, that the user interface for manipulation should be minimal to not challenge the content.

Wand in AR allowing for direct object control with 6DoF

Connected to that, Rodrigues et al. [2017] propose a similar idea of connecting virtual on-screen selection with tangible interface paradigms. Instead of a rotary knob the user controls a so called BatWand, a physical marker which can be moved providing true 6DoF interaction. In contrast to Cannavò et al. [2017], the on-screen menus are minimized and only used to change between general modes as can be seen in figure 2.7 (b). For example, in grabbing mode the selected virtual object translates and rotates like the BatWand. Using a marker to modify virtual objects is also presented by Seidinger and Grubert [2016], who focus on the use case of 3D character customization.

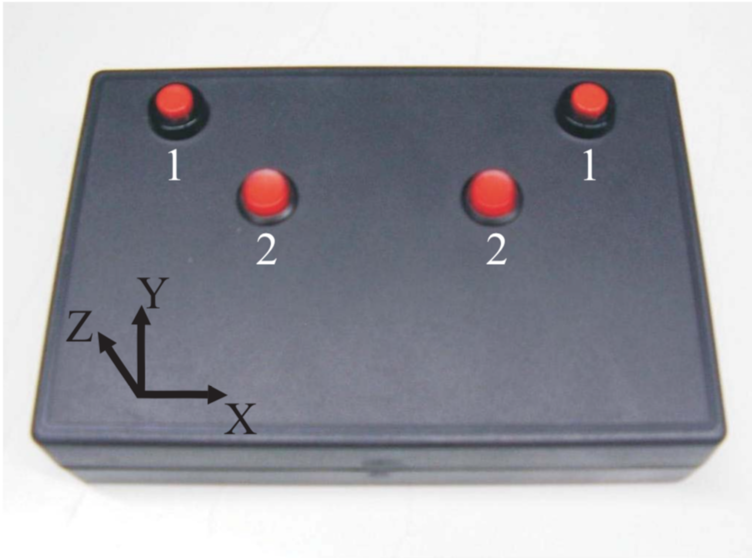


Figure 2.8: Tangible box by Reifinger et al. [2008] with sensors detecting its inclination and buttons for object and mode selection.

Tangible user interface compared to mouse/keyboard and gesture-based interface in AR

Another approach to translate and rotate objects in Augmented Reality is taken by Reifinger et al. [2008]. They compare three interfaces for 3D object manipulation in AR. One being based on mouse and keyboard, another one being gesture-based and finally their tangible user interface, which can be seen in figure 2.8. Internal sensors detect the inclination of the tangible box for translating or rotating a virtual object. Note, that the box works as a relative input device, so there is no one-to-one mapping of tangible box orientation and virtual object orientation, but its inclination of the separate axes is used to manipulate objects. The buttons on the box can be used for object selection, stepping through all objects. Furthermore, they are used to switch between translation and rotation mode. In a study users should translate and rotate objects using all three interfaces. The gesture-based one turned out to be the fastest input for the task, but still being slower than a similar task using real objects in a real environment.

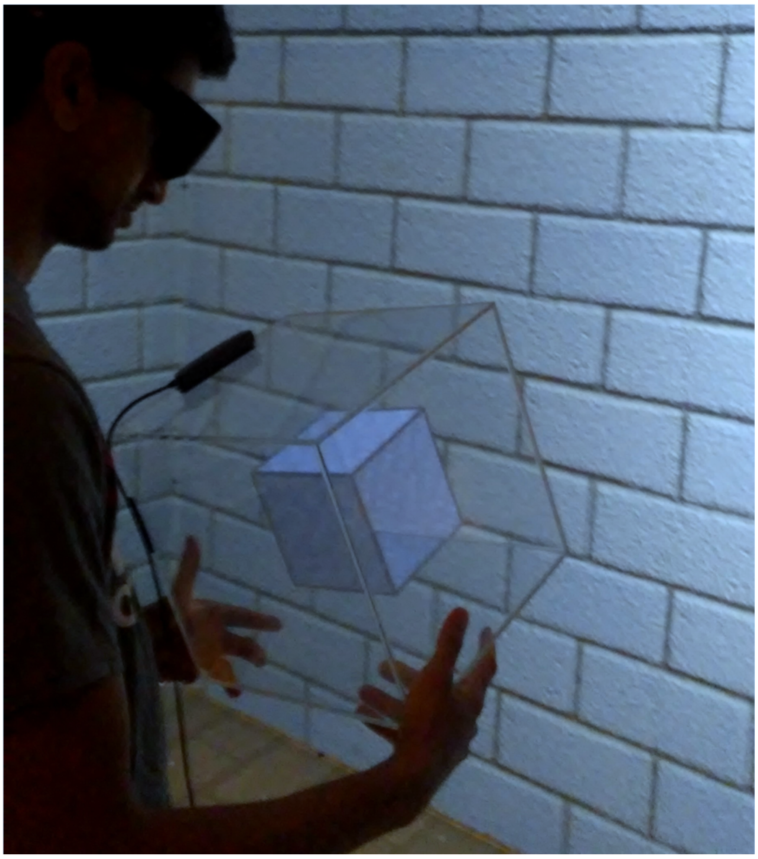


Figure 2.9: Physical Specimen Box by Zielinski et al. [2017] with virtual content.

In VR Zielinski et al. [2017] address the problem of missing haptic feedback by developing a so called Specimen Box. Inside of this physical box virtual content can be positioned. Translating and rotating the actual box can then also manipulate its virtual content. In a user study participants should either use the haptic box or a gesture based grab-and-twirl-method. Overall, the box outperformed the version without haptic feedback. It also led to less rotation per second when using it, which could indicate that the user has a higher understanding of the spatial orientation using the box and can naturally minimize the required rotation.

Use trackable tangible box in VR to provide haptic feedback in otherwise virtual environment

2.2 Tangibles

Benefits of tangible user interfaces

Examples like the rotary knob and wand in AR or the Specimen Box in VR have shown, that tangible user interfaces are already considered an option to control virtual objects. Embedded in virtual content, they can provide haptic feedback that otherwise would be missing. This can benefit eyes-free operation as Weiss et al. [2009] propose. Other already evaluated properties of tangible user interfaces are presented in this chapter.

Tangibles with special purpose can be superior to single general purpose device

Fitzmaurice and Buxton [1997] show that using tangibles in a space multiplex scenario can be beneficial to using a single general-purpose device (e.g. mouse) in a time multiplex scenario. As a conclusion, they suggest that using special devices for specific tasks can be superior which contradicts to the permanent usage of keyboard and mouse as input. This supports our approach of designing a specific input device for object manipulation in 3D. Connected to their final statement *"The ultimate benefit may be to have a collection of strong specific devices creating a strong general system"*, we are going to create a special input device for the rotation task, while still embedding it into the general tangible framework.

Tangibles increase performance when being used for rotation input

Voelker et al. [2015b] investigate the influence of tangibles on one-dimensional rotation tasks around the z-axis. They use tangibles placed on a touchscreen as rotary knobs and compare this tangible user interface to an all virtual one using direct touch input. Two versions of tangibles are tested: round tangible pucks (figure 2.10 (e, f)) whose orientation on screen defines the input, and tangible knobs (figure 2.10 (b, c)) consisting of two parts, where the upper part can be rotated in relation to the fixed lower part. These are compared to one-touch (figure 2.10 (d)) input, where the absolute position of one finger defines the input, as well as two-touch (figure 2.10 (a)) input, where the relative input of the two fingers defines the rotation. In a study participants should rotate from some idle state to a target value with one of the four input methods. Of special interest in the comparison between virtual and physical input is, how it can be performed eyes-on versus eyes-free, as tangibles could

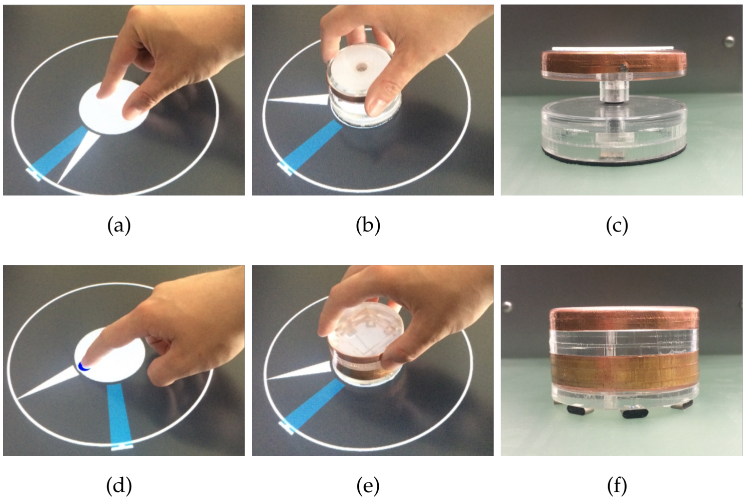


Figure 2.10: One- (d) and two-touch (a) virtual knob compared to tangible knob (b, c) and tangible puck (e, f) (Voelker et al. [2015b]).

potentially give more distinct haptic feedback. Therefore, the tasks should either be performed eyes-on with direct feedback around the object, eyes-free with feedback on a separate display without direct sight on the input, and peripheral, where the feedback can be seen, but is not in focus. As a result, participants overall performed faster using tangibles and yielded less overshoots than when using one-touch input. Especially one-touch input performed worse in the eyes-free task, indicating the missing feedback. Overall this study also motivates our work of using tangibles for the whole 3D object rotation task. They could allow for precise and fast input of rotation data not only around the z-axis.

Tangible equipped with additional relative trackball

Hancock et al. [2009] use tangibles not only for one-dimensional rotation input, but for 3D object manipulation in a different way than we propose. They construct a so-called TableBall, which is a combination of tangible and trackball. The whole device can be moved on-surface, the attached trackball can provide additional relative input. In a user study this variant was preferred by participants over touch when exploring data.

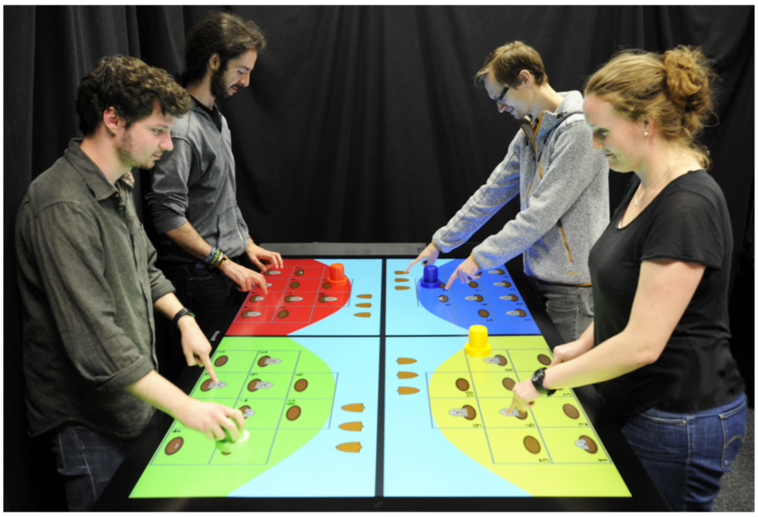


Figure 2.11: Large tabletops allow for collaborative work, tangibles can increase awareness of each other's actions (Cherek et al. [2018]).

Tangibles increase awareness of each other's actions on large tabletops

Today's large multi-touch tabletops allow to work together collaboratively with a personal working space assigned to each user. A study by Cherek et al. [2018] evaluates the influence of tangibles on the awareness of each other's actions in this personal workspace. As can be seen in figure 2.11, two to four players play a collaborative version of Whac-A-Mole using touch with or without additional tangibles. Each player has its own part of the screen where all game actions are executed. Some of these actions are attacking moves against other players, which require an active reaction from their side. The reaction time can be measured and analyzed. As a result, users reacted significantly faster to other's attacks when using tangibles, which indicates increased awareness. Connecting that topic to our work, it would be interesting to see, if awareness even increases when using the tangibles not only on the screen but also in midair.

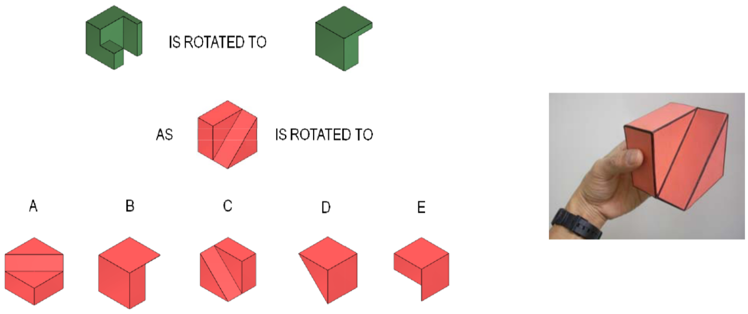


Figure 2.12: Tangibles with a gyroscope can be used to improve students' spatial abilities (Ha and Fang [2013]).

Use tangibles to improve spatial abilities of students

Ha and Fang [2013] equip tangibles with a gyroscope, aiming at improving students' spatial abilities. Using their tangibles, the orientation of the physical object can be detected and subsequently mirrored to a virtual object on a display. Students therefore can physically execute rotation tasks and see the virtual result in real time, instead of just mentally imagining both. This could improve their abilities for tasks like shown in figure 2.12. As a restriction, their tangibles only work for this special purpose and cannot be detected by a touchscreen to create input on it as well. Furthermore, they only describe their prototype building process and do not evaluate it.

Framework to develop educational applications including tangibles

To ease the development of educational content for multi-touch tables using tangibles, Ehlenz et al. [2018] describe their work on a supporting framework. It allows to develop for multiple devices, can be used to collect learning analytics and supports the integration for tangibles.

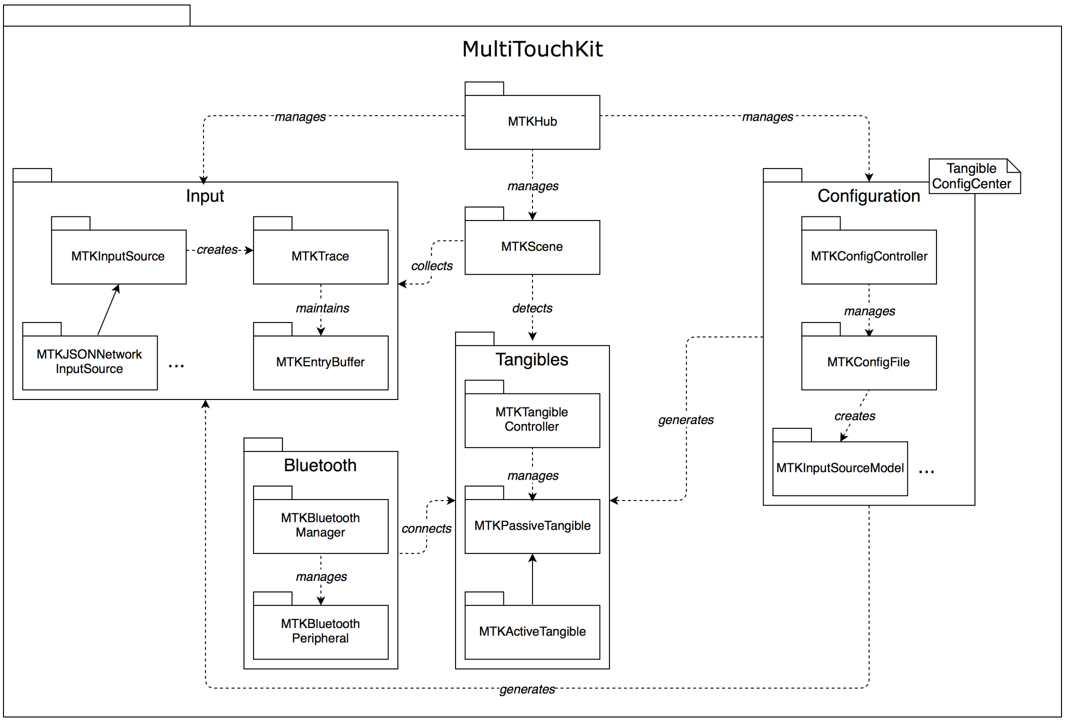


Figure 2.13: Package diagram to visualize the general framework structure.

2.3 Framework

MultiTouchKit to process input and manage tangibles

The framework used for the tangible user interface in this and other projects is called MultiTouchKit. As the name already suggests, its general task is to receive and process touch input. Along with that, it stores and manages tangible properties and consequently detects the position of tangibles placed on a multi-touch screen. It is based on SpriteKit and extends its functionalities to also offer multi-touch support on the Mac. Recently, the framework has been rewritten in Swift to be future-proof. The general structure of the framework is presented in figure 2.13. In this chapter, it will be taken a closer look at its update loop and scene management to understand where the rendering of virtual 3D objects can be included. In addition to that, its tangible management is examined with a closer look at the communication protocol, as this has to be extended for the new off-surface tangibles as well.

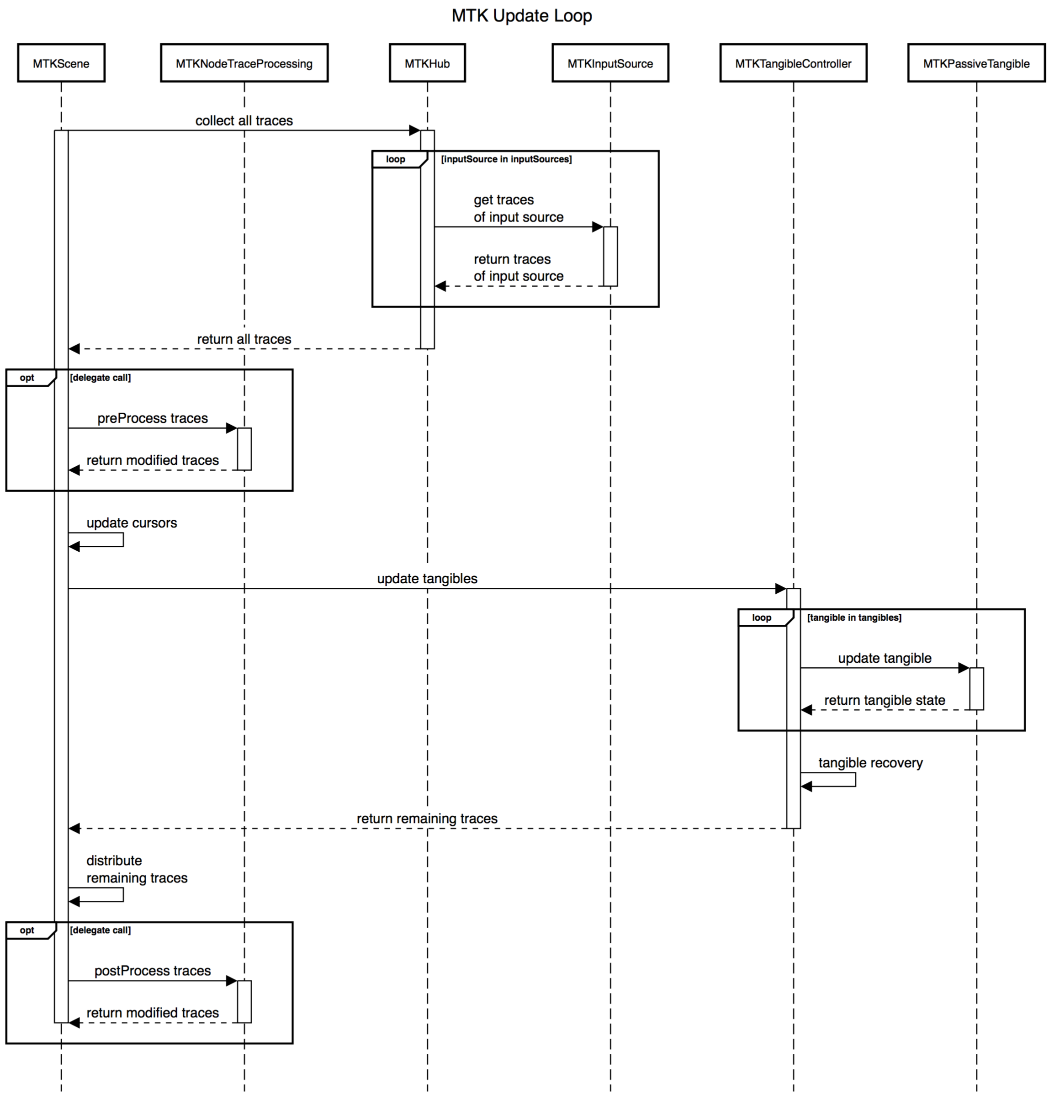


Figure 2.14: Sequence diagram to visualize the performed update sequence.

2.3.1 Scene Management

As the MultiTouchKit is based on SpriteKit, it also uses its concept of scenes. The so-called `MTKScene` provides a 2D (x,y)-coordinate system along with a scene graph consisting of nodes. The scene graph determines the spatial and logical relationships of all elements rendered on screen. Therefore, every object of the scene including tangible instances are basically nodes arranged in this scene graph.

MTKScene with scene graph arranging all elements in nodes

Input sources create MTKTrace instances

Input can be received from multiple sources. At this point, the framework supports input via mouse and network. While the mouse input is mainly used for debugging, the network source is required to get touch events sent from the Microsoft Surface Hub working as our tabletop screen. Other input sources can be added modularly, every input source has to create instances of `MTKTrace`. A trace represents the lifetime of one continuing input, e.g. a touch. It holds information about its state (beginning, moving, or ending) and maintains a `MTKEntry` buffer to save its position for each frame. Gesture recognizers could for example use this data to detect certain movements.

MTKScene collects input and distributes it to tangibles and UI

Figure 2.14 shows the sequence which the framework follows each frame. First, all active input sources manage their input by creating and updating their traces. All traces of the scene are collected in a set which can then be optionally manipulated by a delegate in the `preProcess` call before anything else is done with the input. After that, cursors are updated to the new positions. Next, the traces are associated to tangibles if possible. Existing tangibles are asked to update their status. In case some traces are lost, they try to recover searching for appropriate candidates in the set of free traces. Subsequently, all traces which are not bound to a tangible are associated with other nodes in the scene graph, e.g. UI elements. Finally, the `postProcess` delegate call allows for altering traces again if wanted.

2.3.2 Tangible Management

Passive tangibles are defined by a touch pattern of three points

Tangibles in the framework exist in two forms: passive and active. Passive PUCs, originally developed by Voelker et al. [2013], are defined by a touch pattern of three points. In the update loop of each frame this pattern has to be detected in the set of all traces. Matching traces are then bound to the tangible instance. Fully recognized tangibles have a defined position and orientation on screen. If one or even two of their touches are missing, these parameters get imprecise and the tangible changes to the recover state. If no touch is detected at all, the tangible is not recognized. To tell different tangible instances apart, their patterns have to significantly differ from each other. In addition to that, stationary touches are a problem. Created by a tangible staying at a its position, they are filtered out by the screen after some time.

Active tangibles consist of additional sensors to for example detect if they are currently on screen or not

Motivated by these challenges, active tangibles refine the concept of passive tangibles, using additional sensors and a Bluetooth module as described by Voelker et al. [2015a]. The wireless communication with the framework is organized by the `MTKBluetoothManager`. Each active tangible instance is associated with a unique Bluetooth ID, allowing to assign sensor updates to a certain instance. The manager subscribes to the offered services of the module and gets notified if a value has changed. In case of the surface sensor detecting the underlying screen, it allows for distinguishing between being on table and not on table. Synchronized, this information can be used to assign different active tangible instances to identical patterns. If they are placed on the screen with a slight time difference, the timestamp of beginning traces and the status change of the sensor can be matched. This allows for a more robust detection of tangibles, which can even stay stationary.

Chapter 3

Implementation

The evaluation of related work has shown, that diverse approaches to tackle the task of 3D object rotation exist. Especially in combination with virtual content in AR and VR tangible solutions are proposed frequently. This motivates to use object rotation as a sample application to make tangibles usable on-surface and off-surface in midair for the first time, adding a whole new dimension to the interaction. In midair the tangible rotation controls the rotation of the virtual object, on-surface its translation controls the translation of the virtual object underneath. To subsequently evaluate the performance of this approach, it is compared to input via touch with 2DoF and via 3D mouse with 6DoF. How all these input methods are implemented to be used on our tabletop is presented in this chapter.

Overview

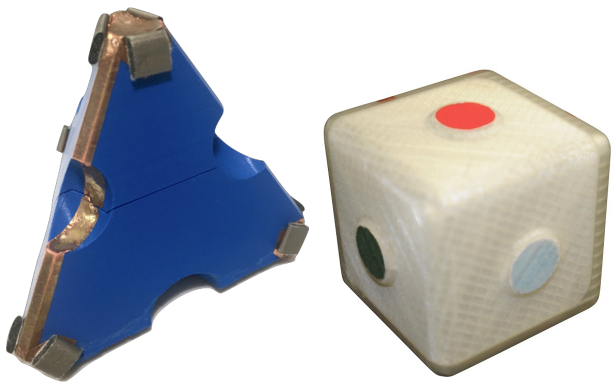
3.1 3D Object Manipulation Using a Tangible

As presented in section 2.3, active tangibles use sensors which provide information to help with tangible detection. Apart from this, the sensors do not serve a purpose yet. Furthermore, the MultiTouchKit is primarily used to render content in 2D with SpriteKit up to this point. Because of that, hardware and software have to be extended to use

Extend tangible hardware and software

tangibles for rotation of 3D objects. First, it is shown how the new tangibles are constructed to track their orientation. Second, it is taken a look at the software. A detailed description of the Bluetooth protocol shows how it is reworked to allow communication in a more modular way coping with different sensors. Moreover, it is shown how to render 3D objects and route input between 2D and 3D using the framework without restricting its existing functionality.

3.1.1 Tangible Hardware



(a) First iteration (b) Second iteration

Figure 3.1: On the left the prototype in a pyramid shape, on the right the second iteration in a cube shape.

Two iterations on tangible design

To track its orientation, the new tangible is equipped with a MPU-6050 tracking device. Combining a gyroscope and an accelerometer, the orientation can be detected and sent out via a Bluetooth module. In the first version, all modules are packed in a compact pyramid shape as can be seen in figure 3.1 (a). Using this shape in practice shows, that an association between pyramid object and each rotation axis is difficult, because not all axes can be orthogonal to one distinct face of the tangible. Consequently, in a second iteration the design is changed to a cube shape, shown in figure 3.1 (b). By that, the tangible faces can be clearly associated with axes. On one face of the cube marker pads are placed to allow detecting the tangible when placed on the tabletop with that side down. They are not placed on every side of the tangible as this could distract the user and weaken the association with the virtual object. Furthermore, the update rate of the sensor is doubled from 10 Hz to 20 Hz in the second iteration to make the interaction more fluent.

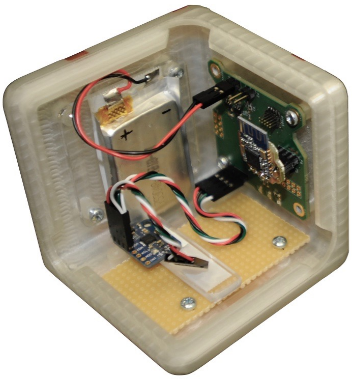


Figure 3.2: The inside of the cube tangible is packed with microcontroller, BLE module, tracking device, and battery pack.

Current hardware restrictions

Figure 3.2 shows the internals of our tangible. As the tangible is not equipped with a magnetometer yet, sensor values slightly drift around the z-axis. While not noticeable over a short time period, the drift aggregates over time, also influencing the spatial position of the other axes. This motivates the implementation of a software recalibration process, which is described in the next chapter. In addition to that, it is worth mentioning that the surface sensor used in PERCs (Voelker et al. [2015b]) is missing in this version, so from a detection viewpoint the tangible works similar to a passive tangible, only tracked by its created pattern. This is sufficient for the study, as the tangible cube is touched by the user permanently and only one is used at a time, but could be changed for future versions.

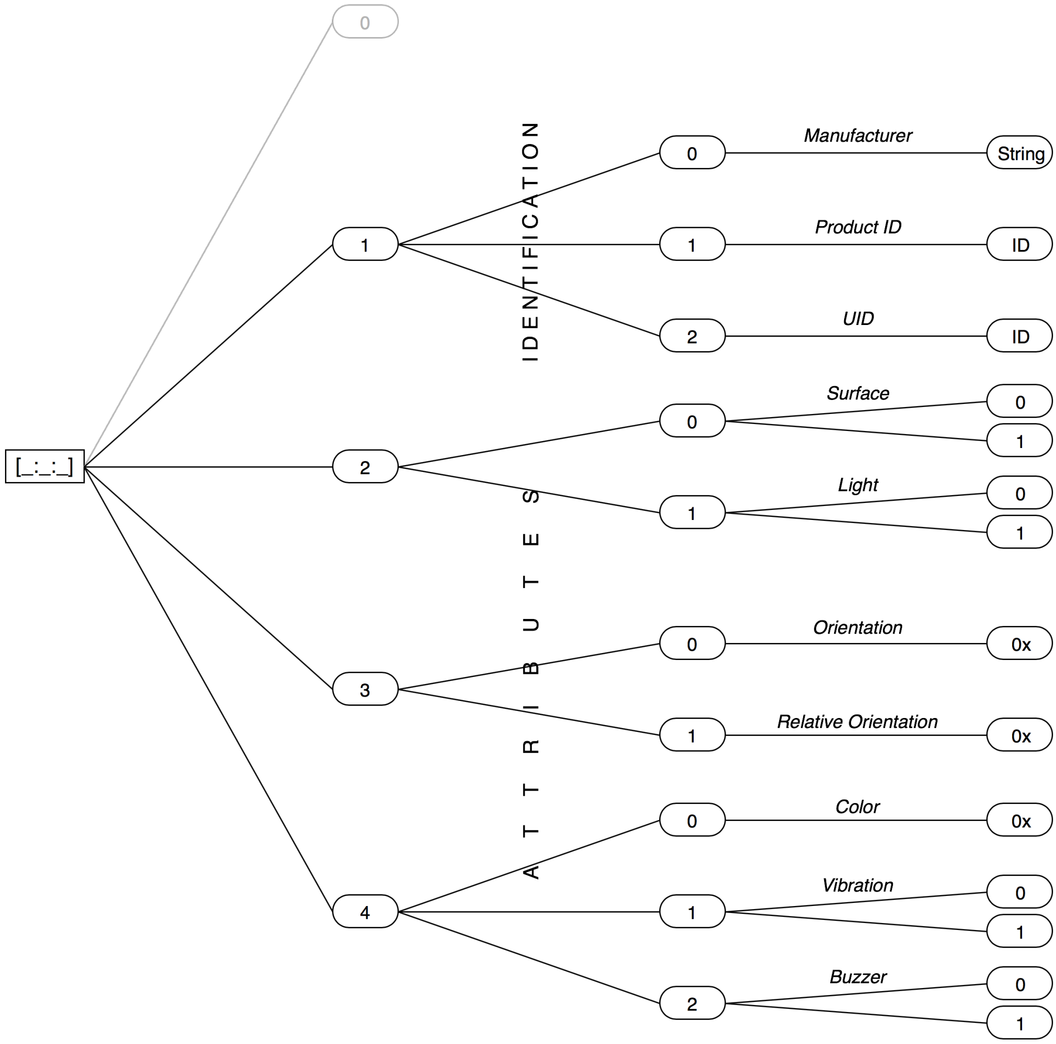


Figure 3.3: Tree structure of the message format used between central and tangible peripherals.

3.1.2 Tangible Software

Tree structure of communication protocol allows for modular data requests

As described, the new tangible is equipped with a tracking device providing information about its orientation to the framework. In addition, other tangible variants could be equipped with different extensions, like a vibration motor or a LED. Furthermore, there are the existing surface and light sensors for tangible detection. To reconcile all these different sensors, a new Bluetooth protocol is used to ex-

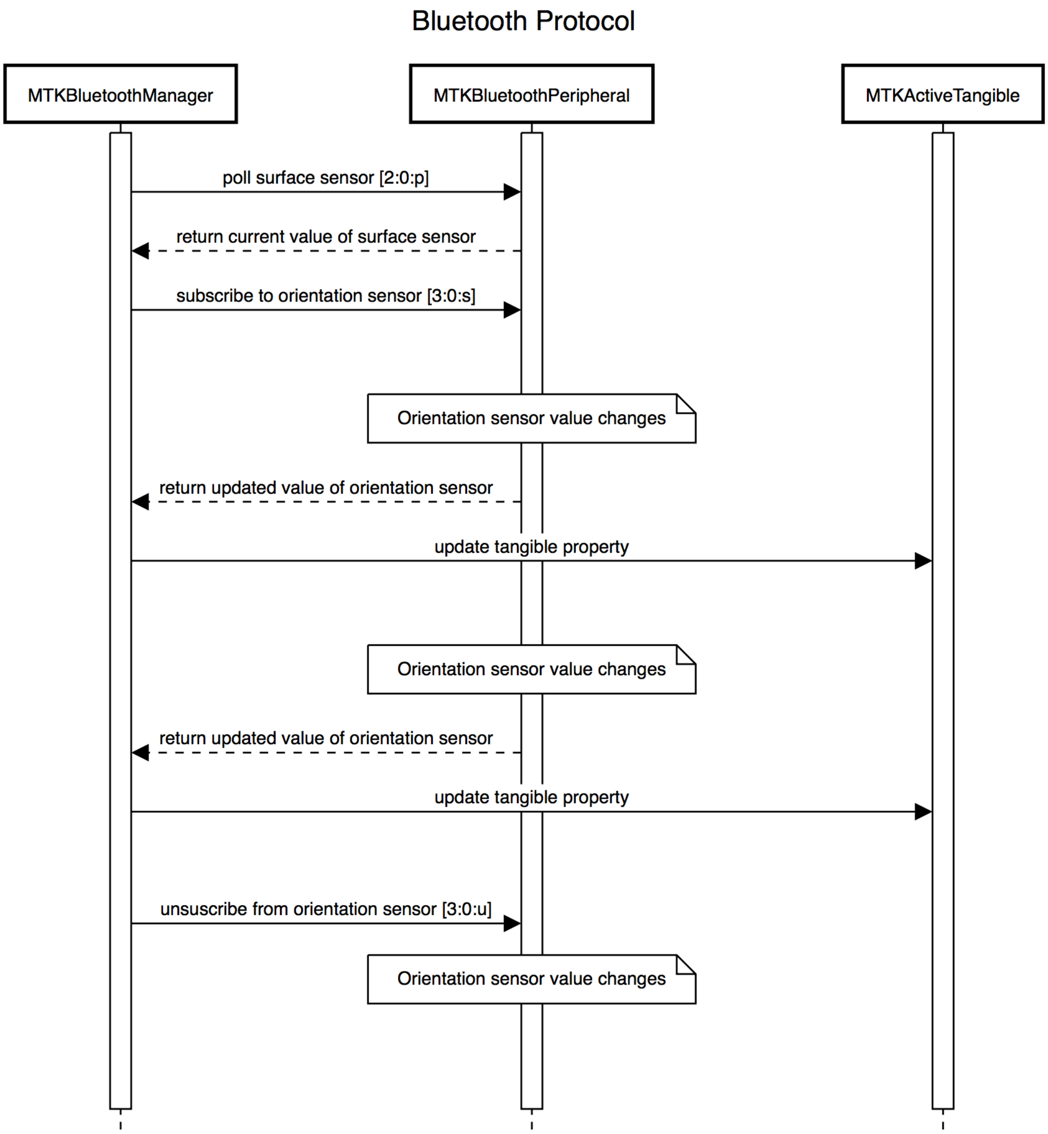


Figure 3.4: Sequence diagram to visualize an example flow of Bluetooth messages.

change messages with the tangibles in a modular way. Figure 3.3 shows, that the message format has a tree structure. Each of the branches specifies another type of information which can be sent to or received from the tangible. This allows for a modular gathering of sensor data.

If the sensor is available, data can either be polled or subscribed to as shown with the sequence diagram in figure 3.4. Polling data means requesting its current value once.

Data can be polled or subscribed to

For updates over a longer period, it can be subscribed to certain sensor values. By that, the framework gets notified as soon as the value has changed. Of course, it can also be unsubscribed from sensors to stop receiving updates. All this is done by the `MTKBluetoothManager`, which then also notifies the associated tangible instance and updates its stored properties. Applications therefore do not have to explicitly deal with that and can simply use these properties to work with. For example, our scene can just use the orientation property of the active tangible. Other types as the LED would also allow to send data to the tangible in the third part of the message. Then it can be specified for example what LED color should be shown or if the vibration motor should be on or off.

Interpret and modify sensor data using quaternions

The sensor data which is sent from the tangible is always defined in relation to a fixed origin. It is set by the tracking device of the tangible when turning it on. Consequently, the tangible sends out its orientation as three Euler angles, whose are then first converted into quaternions. These allow for a seamless calculation of orientation differences especially important to calculate offsets for software calibration, aggregate performed rotation each frame, or check if the target orientation has been reached. For example, given two orientations q_1 and q_2 the difference between them can be calculated by:

$$d_{12} = q_1^{-1} \cdot q_2 \tag{3.1}$$

For an absolute mapping it is important, that the rotation axes always lay in parallel or respectively orthogonal to the tabletop surface. Therefore, the tangible is calibrated at the beginning of its usage on the table. The orientation values that it is sending when placed on the surface being in parallel to the virtual coordinate system are taken as an offset:

$$offset = orientation^{-1} \tag{3.2}$$

This can then be used in the following to calibrate each upcoming orientation in relation to the calibrated position:

$$orientation_{calibrated} = offset \cdot orientation \tag{3.3}$$

To illustrate the progress we can have a look at the moment when the tangible is calibrated. Here, the tangible is at its new idle position and we are multiplying the orientation directly with its inverse. This consequently leads to the identity quaternion for the calibrated orientation in idle position. This calibration process can also be repeated from time to time as it also allows to reset the mentioned drift on the z-axis by software.

Embed 3D content into 2D scene

Up to this point, the orientation data of the tangible can be received and processed. To make use of it in practice, we first have to include 3D content into the MultiTouchKit. SpriteKit allows to integrate 3D content of SceneKit in a so-called `SK3DNode`. This node can be added to the scene graph as usual. It is then associated with a `SCNScene` where content can be placed. The advantage of this method is, that the tangible detection can work on the 2D layer as before. On the other hand, the 3D content has to be managed separately. This is especially important for 3D UI elements, which will be described in chapter 3.3. Following up, technical limitations and solutions we found are presented.

Not using pure SceneKit restricts some options and leads to visualization bugs

Because there is no separate `SCNView` available, more advanced settings like enabling anti-aliasing cannot be defined. Furthermore, we found a bug when visualizing certain 2D shapes above 3D content. If the SpriteKit shape is placed in the z-order above the `SK3DNode`, it is sometimes not shown. Testing the same setting on iOS, the 2D shape is shown when running in the simulator and not shown when running with the same code on an actual device. The issue has also been discussed in support forums, without a solution yet. Luckily, 2D content containing images can be drawn, so it does not restrict our work that much.

Work with quaternions in GLKit

Another general restriction of SceneKit is its limited support for quaternions. These can be assigned to the orientation property of any object, but are basically just stored as vectors. Because of that, we found it more convenient to convert them into GLKit quaternions, use their built-in operations to make advanced calculations, and then convert them back afterwards.

Increase hit test area for an easy translation on-screen

The tangible has to be placed on the object and then moved on the tabletop for translation. With 2D sprites this is a straight forward task as their boundaries remain static, but 3D objects can be rotated in orientations where their exact shapes can be difficult to hit. If the hit test is performed using the actual 3D model, we sometimes found it difficult to place the tangible exactly over it and not on free areas in between. Therefore, we instead place an invisible sprite on the object having the size of its bounding box and use that for the hit test. This makes object translation way more convenient.

Use passive and active tangible instance in parallel to work with new type of tangible

In the past, there always was a clear separation between passive and active tangibles with all active ones providing additional sensor data for tangible detection. With our new 3D tangible this is different. It is active in a way that it connects to the framework vial Bluetooth sending its orientation. But it is passive concerning its detection process, because no additional surface sensor values are sent. Because of that, two instances of the cube tangible are defined in the scene. One active instance is used to connect with the `MTKBluetoothManager` and the rotation off-surface and one passive instance is used to be detected in the set of all touches for translation on-surface. This allows to keep the existing detection algorithms without much effort.

3.2 3D Object Manipulation Using a 3D Mouse

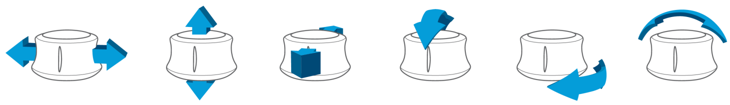


Figure 3.5: Visualization of 6DoF provided by the SpaceMouse as published by 3Dconnexion[1].

SpaceMouse support implemented for comparison with designated hardware used by professional 3D designers

To compare the new tangible with other state-of-the-art input methods, the SpaceMouse by 3Dconnexion is supported to deliver input for object manipulation as well. Its flexible knob provides 6DoF as can be seen in figure 3.5. It can be panned parallel to the surface, lifted and pushed, tilted and last but not least torqued. For the purpose of getting the data from the device drivers, a framework by Martin Majewski is used, which is published under MIT license[2]. Slightly modified, it sends the raw six-dimensional input to the `MTKScene`, which then processes it. Input dimensions are mapped to manipulation axes as 3Dconnexion proposes in their SDK. The driver sends values depended on the driver settings for overall speed, in this context it is left in the default position. Panning parallel to the surface is used for translation. As the object is only translated in two dimensions on the screen, the input created by lifting and pushing the knob is not used. Translation is scaled linearly to allow movement of at most 32 pixels per frame. For rotation in all three dimensions, tilting and torquing of the knob is used. Values are scaled linearly as well, allowing for a rotation of at most 5.625 degree per frame.

[1] www.3dconnexion.com/service/manuals-and-datasheets.html
[2] www.github.com/MartinMajewski/ToolShelf-4-3Dconnexion

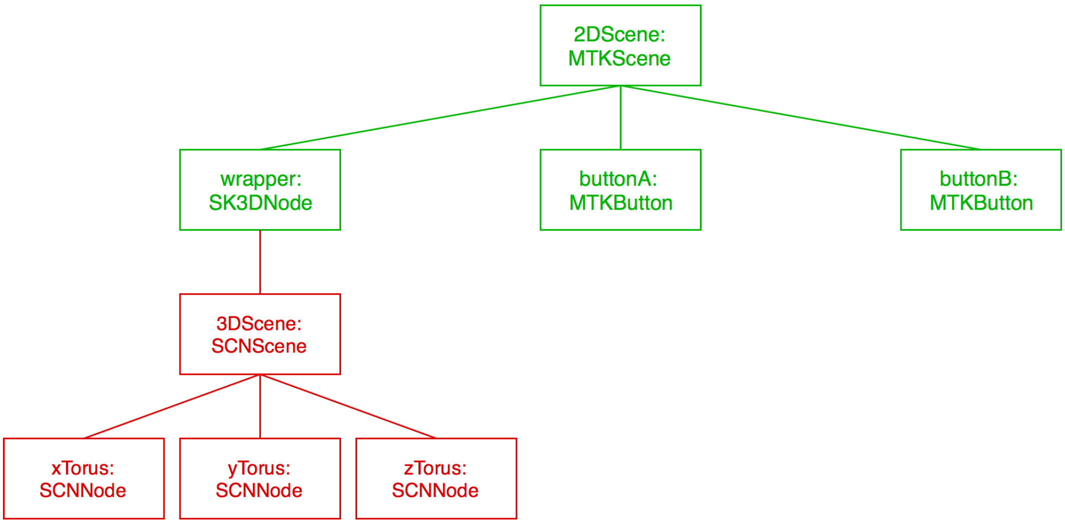


Figure 3.6: Simplified scene graph of a SpriteKit scene in green with an embedded SceneKit scene in red.

3.3 3D Object Manipulation Using Touch

Touch interaction implemented using a rotation gizmo and a virtual trackball

To take a 2DoF input method into account as well, touch input is supported to manipulate virtual objects. Inspired by professional 3D design tools dealing with 2DoF input (e.g. by a conventional computer mouse) a combination of rotation gizmo and virtual trackball is implemented.

Input on 2D layer has to be explicitly converted to 3D

The rotation gizmo requires to embed three-dimensional user interface elements around the object. As figure 3.6 shows, whole SceneKit scenes can be embedded into SpriteKit using a `SK3DNode`. However, this node consequently wraps all 3D content into one image visible to SpriteKit. Therefore, the MultiTouchKit can distinguish between *buttonA* and *buttonB*, but it cannot assign input to specific 3D elements like *xTorus*, *yTorus* and *zTorus* natively. To be capable of that, conversion functions from SceneKit have to be used. They map 2D coordinates given by the SpriteKit scene to 3D coordinates in the SceneKit scene and vice versa. This allows for a precise distribution of touch input to nodes in SceneKit. But it has to be additionally implemented on top of the provided MultiTouchKit functionality.

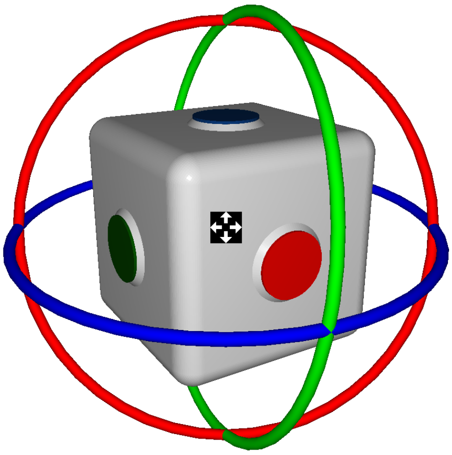


Figure 3.7: Rotation gizmo with three separate axis controls around the virtual object.

Rotation gizmo for separate axis control

With this additional handling, the rotation gizmo as shown in figure 3.7 can be implemented. Selecting one torus enables a rotation around that one axis based on the relative circular movement around the object center. The button in the object center allows for translation.

Virtual trackball to simulate direct 3D rotation with 2D input

Additionally, the virtual trackball method is implemented. It allows for simulated grabbing of the object based on touching the object and the follow-up movement. This is done by translating 2D input to a virtual unit sphere around the object. Given are the two-dimensional input by the user pos_{input}, the virtual object position as the sphere center pos_{object}, and bounds to specify the sphere dimensions $bounds$. First, pos_{input} can then be scaled around pos_{object} with respect to $bounds$ to a still two-dimensional $pos_{scaled} \in [-1, 1]$. The euclidean distance to the origin d of pos_{scaled} indicates if the position is on the unit sphere (≤ 1) or outside (> 1). If inside the sphere, the location on it is defined as:

$$pos_{sphere} = \begin{bmatrix} pos_{scaled_x} \\ pos_{scaled_y} \\ \sqrt{1 - d^2} \end{bmatrix} \tag{3.4}$$

If however outside of the defined sphere around the object, coordinates are set to the borders of the sphere:

$$pos_{sphere} = \begin{bmatrix} \frac{1}{d} \cdot pos_{scaled_x} \\ \frac{1}{d} \cdot pos_{scaled_y} \\ 0 \end{bmatrix} \tag{3.5}$$

As a conclusion, the two-dimensional pos_{input} is converted to the three-dimensional pos_{sphere}. Taking two 3D coordinates, the object can now be rotated in 3D following the touch movement.

3.4 Demo Mode

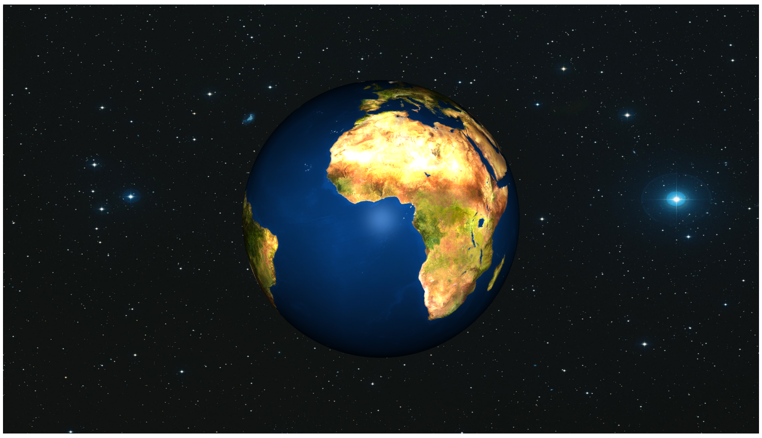


Figure 3.8: The earth model can be used to explore the different input methods.

To try these input methods out, a demo mode is implemented. In this demo the user can translate and rotate a model of the earth using one of the implemented input methods. On figure 3.8 the model can be seen using high-resolution pictures provided by the NASA[2]. In contrast to the study, described in the upcoming chapter, here the user can explore one detailed object without a specific task.

[2] `www.visibleearth.nasa.gov`

Chapter 4

Evaluation

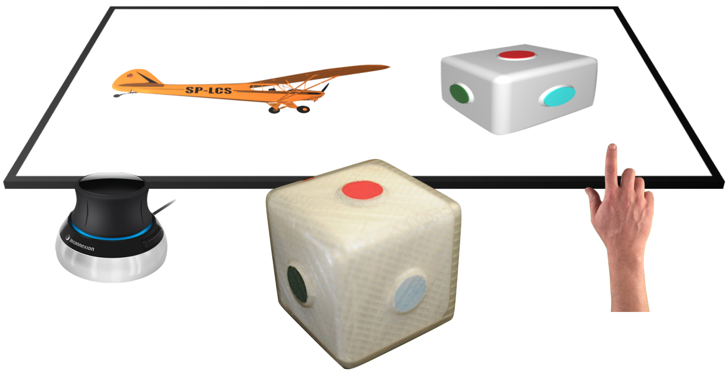


Figure 4.1: 3D mouse (left), tangible (center) and touch (right) are compared to each other providing input for object rotation in 3D and translation in 2D on a tabletop.

Overview

After implementation, the different input methods for 3D object manipulation shown in figure 4.1 can be compared to each other in a user study. To understand how off-surface and on-surface tangible usage work together, rotation tasks are combined with translation tasks. We want to analyze how participants perform using the different input methods and which tactics they develop. To be capable of extracting this information, it will be measured how long participants take to finish object manipulation tasks and how they translate and rotate the controlled object. Subsequently, we present results of this logging, additional user ratings, and qualitative feedback and discuss its implications for each input method. As a conclusion, this analysis will allow us to derive design recommendations on how to use tangibles off- and on-surface.

4.1 Hypotheses

H1: Using tangible input leads to significantly shorter task completion times compared to using touch.

H2: A perceptual analogy between the tangible input control and the controlled virtual object leads to significantly shorter task completion times compared to only a spatial analogy.

H3: Using tangible input without toggle to fix orientation makes translation corrections on-screen more difficult to perform compared to all other evaluated input methods.

4.2 Experimental Design

Pilot study to tweak parameters, visualizations and logging

Environment, input conditions, procedure and measurements were tested in a pilot study with 3 participants. In doing so, we could tweak parameters and optimize visualizations before starting the actual study. This was especially important to estimate how many total tasks a participant could approximately perform in a time frame of at most 45 minutes. Additionally, we could improve the indication of the goal orientation and optimize the logging format for an easier parsing afterwards. In the following sections, the setup and conditions for the final user study are described.

4.2.1 Environment

The tasks were performed on a Microsoft Surface Hub 84′ positioned horizontally as a tabletop. Its display with dimensions of 220x117 cm and a resolution of 3840x2160 pixels can detect up to 100 touch points. In our setup it was fixed at a height of 74 cm. The software was executed on an iMac Pro running with 30 fps on the screen in full resolution. The display was placed at a shared project space, but separated with curtains. Touch could be directly executed on the table without additional hardware. The 3D mouse condition used the SpaceMouse Compact by 3Dconnexion[1]. For both tangible conditions the newly developed tangible, described in section 3.1.1, was used.

4.2.2 Input Conditions

Four different input modes were included in the study:

- *Touch*: combination of virtual trackball technique on the object and rotation gizmo for separate axis control around the object, translation possible via button in the center of the object (details in section 3.3).
- *Mouse*: SpaceMouse with mapping of input dimensions attached to single knob (tilting, torquing, shifting parallel to surface) to object rotation and translation dimensions (details in section 3.2).
- *Tangible*: orientation of tangible continuously mirrored to virtual object orientation, translation by placing tangible on the object and dragging it on-surface (details in section 3.1).
- *Toggled*: orientation of tangible only mirrored to virtual object orientation when toggled, translation by placing tangible on the object and dragging it on-surface (details in section 3.1).

[1] www.3dconnexion.com/spacemouse_compact/en/

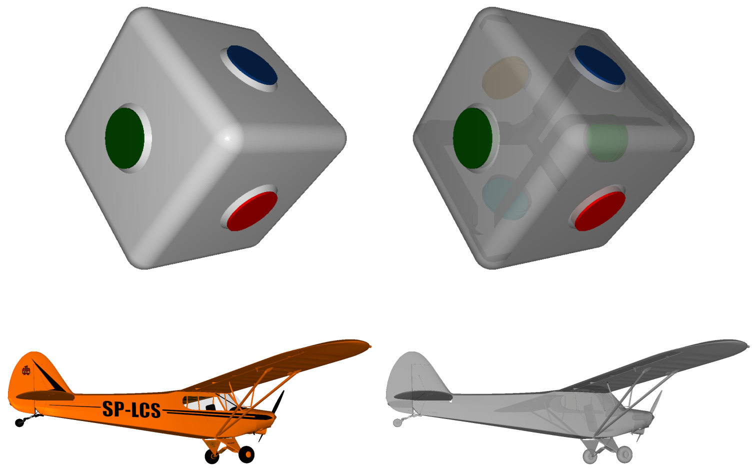


Figure 4.2: Cube (top) and plane (bottom) each with their goal indicator on the right.

Each of these input methods was used to control the translation and rotation of two virtual object types:

- *Cube*: 3D-model of actual tangible as can be seen in figure 4.2 (a), with matching shape and colors.
- *Plane*: 3D-model of plane as can be seen in figure 4.2 (b), with shape and colors distinct from the tangible.

4.2.3 Procedure

1	2	8	3	7	4	6	5
2	3	1	4	8	5	7	6
3	4	2	5	1	6	8	7
4	5	3	6	2	7	1	8
5	6	4	7	3	8	2	1
6	7	5	8	4	1	3	2
7	8	6	1	5	2	4	3
8	1	7	2	6	3	5	4

Table 4.1: Latin square to determine the order in which the different conditions are tested.

General procedure of the study

Consequently, these input conditions led to eight combinations as independent variables: *Touch* + *Cube* (1), *Touch* + *Plane* (2), *Mouse* + *Cube* (3), *Mouse* + *Plane* (4), *Tangible* + *Cube* (5), *Tangible* + *Plane* (6), *Toggled* + *Cube* (7), and finally *Toggled* + *Plane* (8). Participants performed all combinations in a within-subjects design. The order was randomized using a Latin square (see table 4.1) to counterbalance learning and fatigue effects.

Demo of each new input mode to understand basic controls

Every new input mode was first presented to the participant by the instructor. It was demonstrated how each translation and rotation axis can be altered with the designated input method. Afterwards, the participants could move a simple demo object (distinct to cube and plane) and try out the input method by themselves. In total, they could practice until completing two subtasks. Finally, they were asked if they understood the basic principle on how to use the input and if positive it was proceeded with the actual task.

One task divided into six subtasks with one target each

Each task as a combination of input method and controlled object furthermore consisted of six subtasks to perform. During a subtask one target had to be hit. Always starting from an idle position, the object had to be translated to a certain position and rotated to a certain orientation indicated by a goal object. As already shown in figure 4.2, this goal object is a copy of the controlled object without coloring (except the color circles on the cube) and slightly transparent to not occlude the controlled object. We defined six positions and six orientations to provide variety. Additionally, these orientations were slightly varied for each task in a range of $\pm$ 10 degree on each axis. Subsequently, positions and orientations were pseudo-randomized in their order.

Tolerances for reaching a target

To fulfill the subtask, the controlled object had to in a tolerated range around the position and orientation of the target. Tolerances were set to at most 25 pixel ($\approx$ 1,44 cm on the Surface Hub) on each axis for translation and at most 7.5 degree difference on each axis for the orientation. Furthermore, the object had to stay in this tolerated range for at least one second. This additional time frame was defined to prevent lucky hits when randomly moving the object fast in all dimensions.

Breaks in between subtasks possible

In between the subtasks, participants could rest and actively start the next subtask when ready. For both tangible conditions, this break was additionally used to recalibrated the tangible. This was performed in a short procedure by positioning the tangible on-surface with a prescribed orientation on a designated area. By this means, we could reset the described rotation drift which would otherwise aggregate over longer time periods.

4.2.4 Measurements

Logging in the background includes completion times and information about translation and rotation

Time was measured for each target from pro-actively starting the subtask by pressing a button until finishing it by fulfilling the given task. Additionally, translation and rotation delta to the goal were logged to get insights about possible different tactics. All translation and rotation operations were also aggregated to get to know how much the object is moved and rotated in total by the user. Rotation data was logged in radian and could afterwards be easily converted into degree. Translation data was logged in pixel. With knowledge about the pixels per inch (PPI) of the used display, this could also be converted into cm afterwards if required.

Before the tasks were performed, we collected demographics and participants were asked to give information about their previous experiences with the different input methods. After the study, they were asked to rate the four used input methods exclusively and answer questions about their experience on a 5-point Likert scale. These questions included how easy translation and rotation corrections could be performed, if the input method was easy to understand, fast and accurate, and if they think that input and output were closely coupled. In addition to that, in the end general comments could also be made. The full questionnaire can be seen in the Appendix.

User questionnaire for surveying demographics, prior knowledge, and user ratings

4.3 Participants

24 people (aged from 22 to 30, M = 25.5, SD = 2.1, 1 left handed, 5 female, 1 n.a. gender) participated in the study. Regarding experience with input tools, 13 had at least some experience with 3D-design tools. Fusion360, OpenSCAD, 123Design, Maya, Blender, Unity and Cinema4D were mentioned by more than one participant as tools they have already used. 4 stated at least some experience with a 3D mouse and 18 noted that they have at least some experience with tangible user interfaces.

4.4 Results

Key figures

In total, the 24 participants performed 1152 subtasks. Logging their interaction each frame, this resulted in 737745 log entries. Because participants could not finish the subtask without further instruction in nine cases, these were excluded from the data analysis leading to a total of 1143 subtasks to analyze. In addition to that, 576 detail questions were answered and many comments made. In this section, the results are presented.

4.4.1 Quantitative

Data distribution

The data for task completion times and total rotation is not normally distributed. In this case, we therefore perform the analysis on the log-transformed data. In contrast, total translation data is normally distributed. Because of that, here the analysis can be performed directly on the logged data.

No significant difference between targets

There are no significant effects regarding the different targets. This indicates, that the subtasks were equally difficult to solve. As a consequence, in the following we aggregate results for all different targets.

Completion Time

3D mouse was significantly faster than both tangible modes, which were again significantly faster than touch

The task completion time is analyzed to get insights into the general performance of the participants using the different input modes. First the results combining both object conditions only looking at the input mode are presented. ANOVA reveals a significant effect on the completion time for the different input conditions ($F(3, 1112) = 72.34$, $p < 0.0001$). A pairwise comparison using the the Tukey-HSD test indicates that the *Mouse* condition was significantly faster than all other input conditions ($p < 0.0001$ for *Touch*, *Tangible* and *Toggled*). *Touch* performed worst showing significant differences to all other conditions ($p < 0.0001$ for

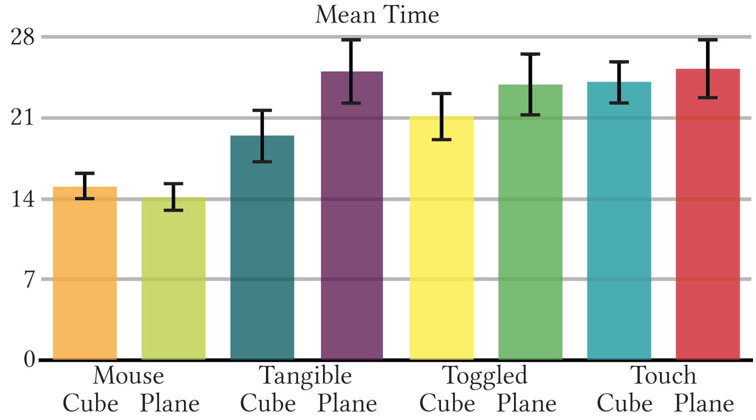


Figure 4.3: Mean completion time (seconds) of all eight tested conditions with 95% confidence intervals.

Mouse and *Tangible*, p = 0.006 for *Toggled*). The two conditions using tangibles *Tangible* and *Toggled* however did not perform significantly different (p = 0.61). Only focusing on the input method, using the 3D mouse on average was 10 sec faster than using touch and 7.8 sec faster than using one of the two tangible conditions.

Cube was significantly faster than plane only when using tangible in continuous mode

In addition to that, we want to have a look at possible impacts of the controlled object. There is a significant general effect on completion time depending on the controlled object ($F(1, 1112) = 6.60$, $p = 0.010$). Looking at the different input modes in detail reveals, that for the interaction between each input mode and each object a significant main effect ($F(3, 1112) = 7.69$, $p < 0.0001$) can be found. Comparing each input with both objects, for most combinations there are no significant effects between *Cube* and *Plane*, however when using the *Tangible* mode, this is different. Placing the *Cube* object correctly took participants significantly less time than the *Plane* object ($p < 0.0001$). Looking at all combinations, users overall performed fastest with *Mouse* + *Plane* (4) (M = 14.15 sec, SD = 6.9), while *Touch* + *Plane* (2) is the slowest condition(M = 25.24 sec, SD = 15.5). For all conditions, except *Mouse*, the cube was placed faster (5.5 sec for *Tangible*, 2.7 sec for *Toggled*, and 1.2 sec for *Touch*). Figure 4.3 shows the timings for all input and object conditions including their 95 % confidence intervals.

Total Translation

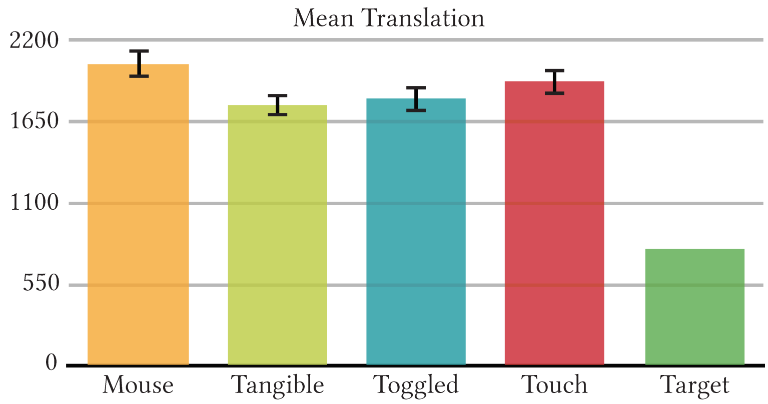


Figure 4.4: Mean total translation (pixel) per input method with 95% confidence intervals and the average target distance.

3D mouse with significantly higher total translation than both tangible modes

To analyze a possible influence of the different input modes on user behavior, the performed translation in pixel is aggregated for each task. A significant main effect on the total translation by the input condition ($F(3, 1112) = 11.95$, $p < 0.0001$) is revealed by an ANOVA. A pairwise comparison with Tukey-HSD reveals, that compared to *Tangible* participants translated the virtual objects significantly farther in both the *Mouse* and the *Touch* condition ($p < 0.0001$ for *Mouse* and $p = 0.0117$ for *Touch*). In addition to that, there is a significant difference between *Mouse* and *Toggled* ($p < 0.0001$), other pairwise comparisons are not significantly different. Because the set of target positions was fixed, as expected there are no significant effects on the total translation by object ($F(1, 1112) = 0.37$, $p = 0.542$). Furthermore, there are no significant effects when taking input and object combinations ($F(3, 1112) = 1.02$, $p = 0.381$) into consideration. The minimal distance players had to move the virtual object directly to the target was 795.8 pixels on average. As figure 4.4 shows, participants overall moved the object more directly to the target position in the *Tangible* and *Toggled* conditions (*Tangible*: $M = 1761.3$ pixel, $SD = 511.1$ and *Toggled* $M = 1804.9$ pixel, $SD = 629.1$). *Touch* ($M = 1920.0$ pixel, $SD = 675.2$) and *Mouse* ($M = 2043.1$ pixel, $SD = 714.9$) needed more transitions to finish a task.

Total Rotation

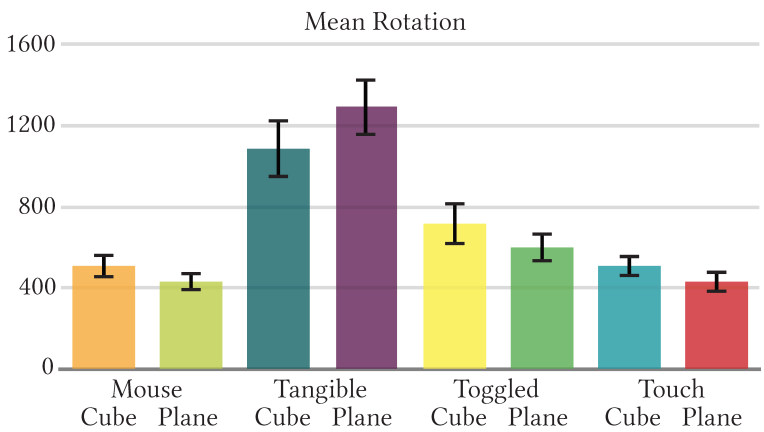


Figure 4.5: Mean total rotation (degree) of all eight tested conditions with 95% confidence intervals.

Both tangible modes with significantly higher total rotation than touch or 3D mouse

One part of the object manipulation task, the total translation, is already analyzed. Consequently, the total rotation is also logged to capture characteristics of the different input mode. An ANOVA shows no significant effect on the total rotation whether *Cube* or *Plane* are used. However, significant differences can be found for the different input conditions ($F(3, 1112) = 169.01$, $p < 0.0001$). A pairwise comparison indicates, that participants rotated the virtual object far more with the *Tangible* or *Toggled* input condition that in the other conditions ($p < 0.0001$ for all comparisons). Additionally, in between the both tangible conditions there is also a significant difference ($p < 0.0001$). On the other side, there is no significant difference between *Touch* and *Mouse* ($p = 0.98$). There also are significant effects on the total rotation ($F = (3, 1112) = 8.79$, $p < 0.0001$) when taking input and object combinations into consideration. For most input modes there is no significant difference between the two objects. Only for the *Tangible* condition participants used significantly more rotation to find the given target when controlling the *Plane* ($p = 0.009$). Figure 4.5 shows the total rotation in degree for all input conditions.

Translation and Rotation Progress

Different progress charts for each input mode reveal characteristics

In addition to all these total measurements for whole tasks, we want to also take a look at the progress during a task. Therefore, we logged the delta between object and goal for both translation and rotation over time. Normalized, this can help to get insights into different strategies users developed with different input modes. Figure 4.6 shows the progress for *Touch*, almost linearly decreasing for both translation and rotation. In figure 4.7 the progress for *Mouse* can be seen, showing that translation and rotation at the beginning decrease faster, but then slower towards the end. Next, figure 4.8 reveals that in the *Tangible* condition first the translation and second the rotation decreases. This progress can also be detected for the *Toggled* condition, but not that pronounced as figure 4.9 shows. In contrast to *Touch* and *Mouse* both tangible modes have local minima in their rotation progress.

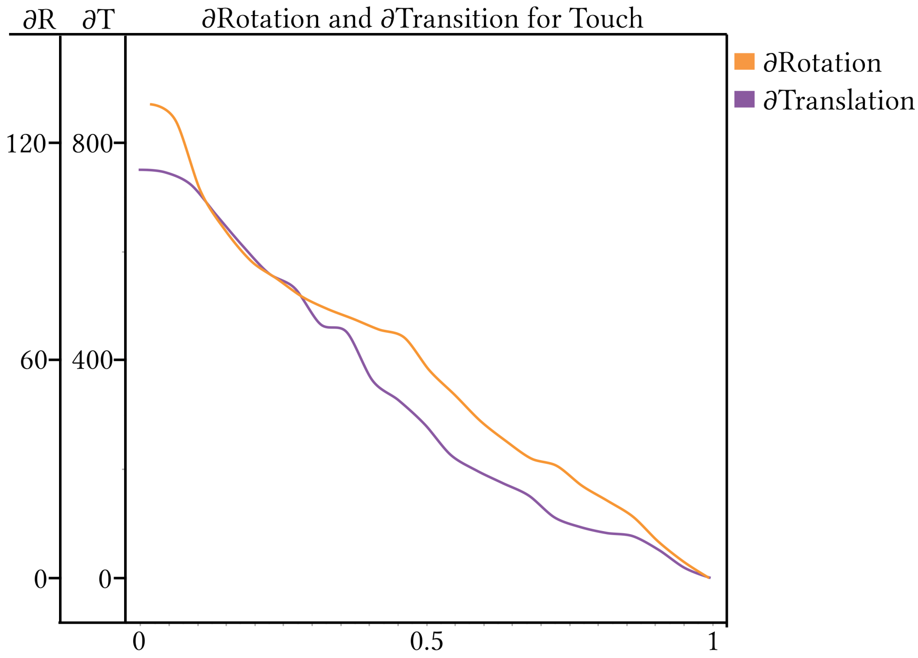


Figure 4.6: Touch input progress to goal over normalized timeline.

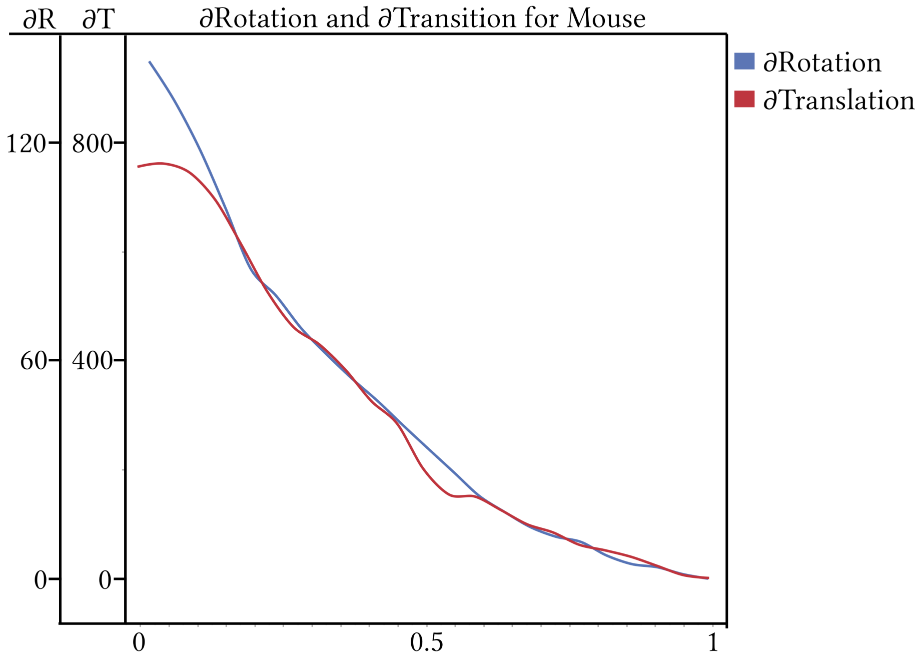


Figure 4.7: 3D mouse input progress to goal over normalized timeline.

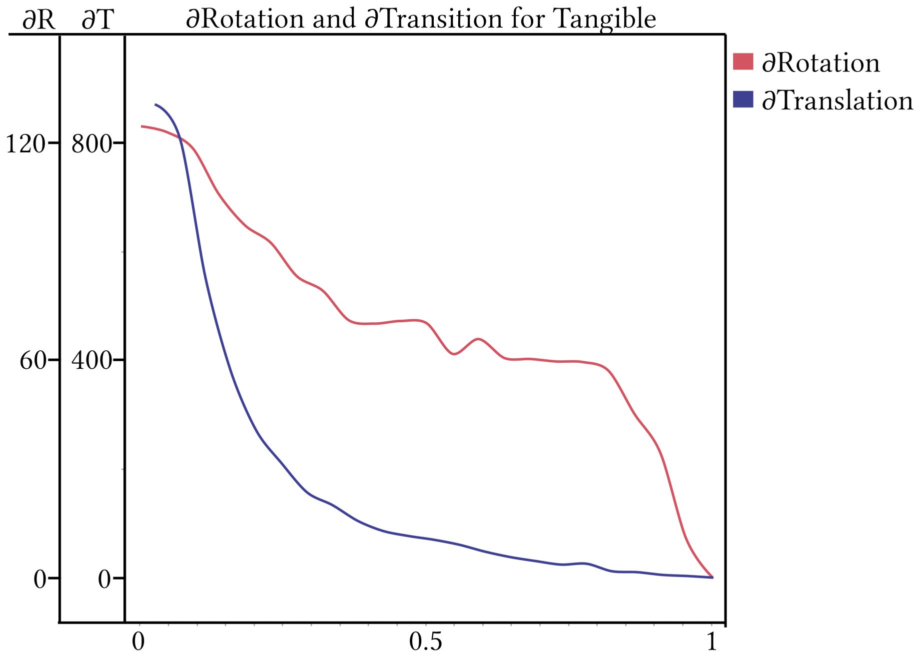


Figure 4.8: Tangible input progress to goal over normalized timeline.

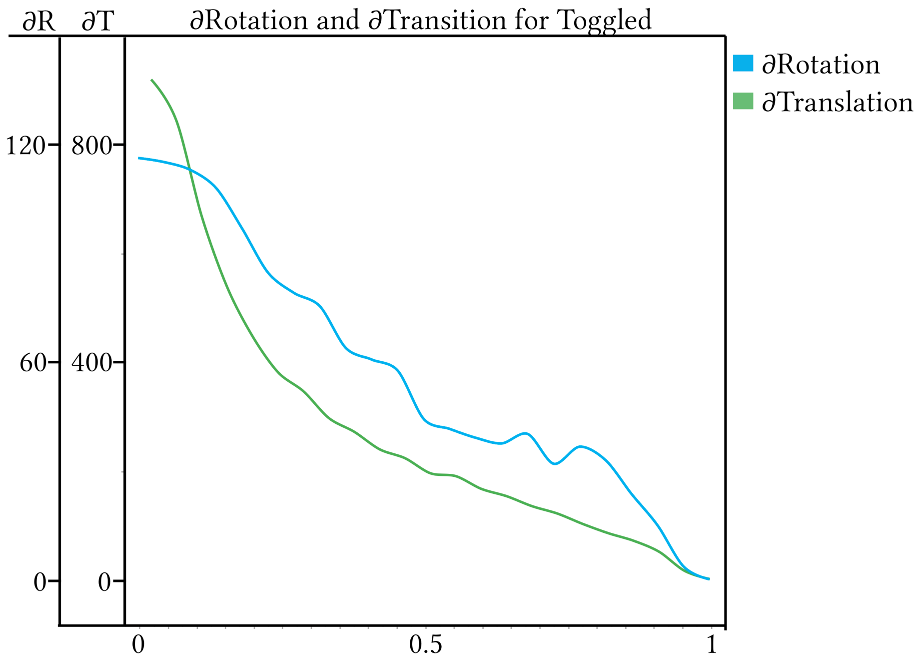


Figure 4.9: Toggled input progress to goal over normalized timeline.

User Rating and Questionnaire

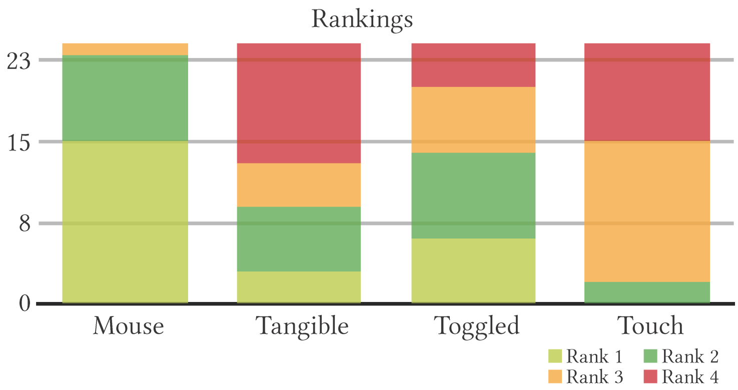


Figure 4.10: User rating for the different input modes exclusively from 1 (best) to 4 (worst).

In addition to the logged data in the background, participants were asked to rate the four input methods exclusively from 1 (= worst) to 4 (= best). Results can be seen in figure 4.10 and show that 15 people rated the *Mouse* condition best, 8 second-best and nobody worst. The TUI however was rated best 3 times in continuous *Tangible* mode and 6 times in *Toggled* mode. On the other hand, 12 people rated it worst in *Tangible* mode and 4 in *Toggled* mode. *Touch* was not preferred by anyone, 2 rated it on the second position, but 9 also rated it worst.

Most people rated the 3D mouse best, other preferences shared between both tangible conditions

	Touch		Mouse		Tangible		Toggled	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Translation corrections were easy to perform	4.04	1.16	3.58	1.25	3.00	1.29	4.00	1.29
Rotation corrections were easy to perform	2.91	0.93	4.46	0.88	4.29	1.04	4.17	1.05
The input method was easy to understand	3.83	1.43	3.75	0.99	4.38	1.06	4.33	1.13
The interaction was fast	2.58	0.93	4.38	0.97	3.38	1.28	3.54	1.18
The interaction was accurate	3.83	1.05	4.25	0.90	3.88	1.12	4.08	0.97
Input and output were closely coupled	3.46	1.28	3.92	1.06	4.13	1.15	3.96	0.91

Table 4.2: Means and standard deviations of questions about each input mode (5-point Likert scale, 1 = totally disagree, 5 = totally agree).

Translation corrections most difficult to perform with tangible in continuous mode

Table 4.2 shows the results for the different questions. Regarding to that, translation corrections were most easy to perform with both *Touch* (M = 4.04, SD = 1.16) and *Toggled* (M = 4.00, SD = 1.29) and were particular difficult in *Tangible* mode (M = 3.00, SD = 1.29). Rotation corrections however were most difficult to perform with *Touch* (M = 2.91, SD = 0.93), and easy to perform in the other modes. Both tangible modes were particular easy to understand (*Tangible* with M = 4.38, SD = 1.06 and *Toggled* with M = 4.33, SD = 1.13). Being closely related to the measurements we presented in the last section, interaction felt fastest with *Mouse* (M = 4.38, SD = 0.97), then both TUI conditions (*Tangible* with M = 3.38, SD = 1.28 and *Toggled* with M = 3.54, SD = 3.54), and worst rated with *Touch* (M = 2.58, SD = 0.93). Interaction felt most accurate with the *Mouse* (M = 3.83, SD = 1.05), with the other modes not far behind. Input and output felt most closely coupled with the *Tangible* (M = 4.13, SD = 1.15), but *Toggled* (M = 3.96, SD = 0,91) and *Mouse* (M = 3.92, SD = 1.06) close behind. With *Touch* (M = 3.46, SD = 1.28), input and output felt not that closely coupled.

4.4.2 Qualitative

Besides the quantitative feedback, participants could also actively write down comments about their experience. In addition to that, the instructor also noted down comments made during the study.

Related to the touch input mode, 7 participants mentioned, that they mostly used the virtual trackball and not the separate rotation gizmo elements for each global axis.

Virtual trackball in touch mode preferred

Concerning the 3D mouse most of the users did not have prior experience using it and during the demo task many stated that they needed time figuring out how to use it. But after this initial phase, they could cope with it well, as 13 out of the 24 total users explicitly noted down that they noticed a steep learning curve with the device. However, 8 participants also explicitly stated, that they found it difficult to alter only a single input axis or make small adjustments.

Learning curve while using 3D mouse

Comments about the tangible experience can also be classified. 5 users stated, that they would like to use their second hand for translation on-screen while performing the rotation in midair. 7 complained about the reset of the orientation once the toggle button was pressed, 5 of them explicitly suggested to use relative instead of absolute adjustments in this mode. Besides that, users liked the tangible mode, it was most fun to use (3 comments), provided haptic feedback (2 comments), and the tangible could be used as reference when controlling the cube (1 comment).

Two-handed mode and relative adjustments could be an option

4.5 Discussion

In this section all quantitative and qualitative results regarding each input method are summarized and discussed. Consequently, their individual interaction characteristics are identified and compared to each other.

4.5.1 Touch

2DoF cannot be directly mapped to 5DoF

The touch method was originally implemented to also examine user performance only providing input with 2DoF. We expected that the transfer from two input dimensions to five output dimensions would influence the performance and preference of participants. Supporting this assumption, touch overall performed significantly slowest concerning task completion time. In addition to that, it got the worst user ratings, not being preferred by anyone.

Touch steady but slow with virtual trackball preferred over single axis rotation

To get a deeper understanding why this could be the case, we can have a look at the progress chart 4.6 of delta translation and rotation between controlled object and goal over time. It reveals that touch provided a steady but slow input reaching the target values with an almost linear decrease in both delta translation and rotation. In combination with the significantly lower aggregated total rotation compared to both tangible modes, this indicates that the input method did not encourage quick rotation movements. Taking the comments into consideration as well it can be said, that the single rotation gizmo elements, which would allow for a fast rotation around a single axis, were not used frequently. We assume that this is because they would require a more abstract thinking, splitting the overall rotation into several rotations around each global axis. When not consciously used, the single axis control led to unwanted behavior. Instead, the virtual trackball method was preferred by most of the participants, due to its comparatively direct way of interaction. It has to be said, that this however did come with the price of reduced speed as it takes more time fully rotating an object taking only the virtual trackball.

Having a look at the detailed user ratings reveals, that this can not only be derived by objective data analysis, but also by subjective user perception. Touch input did not feel fast compared to other modes. Furthermore, comparatively bad ratings for the rotation corrections show that they were identified as a problem and not the translation corrections. As a conclusion, all this indicates that touch input with 2DoF is not supporting an understandable direct and fast way of manipulating objects in all three dimensions.

Especially rotation corrections were not straightforward to perform with 2DoF

4.5.2 3D Mouse

The 3D mouse is frequently used by professionals in 3D graphic design. Therefore, we expected that it could lead to a fast performance, but also wanted to check if this in also valid for novel users without much practice. Overall, the tracked performance gain compared to 2DoF input with touch was tremendous. In the study, it was the significantly fastest method and was preferred by the majority.

Designated hardware providing 6DoF

We want to take a look at where exactly the strengths and weaknesses of this type of input lie. The progress chart 4.7 reveals, that delta translation and rotation to the goal almost decrease in parallel. Combined with our observations during the study it can be said, that users often altered rotation and translation simultaneously to reach the goal. In contrast to the touch method, the 3D mouse allowed for very quick changes of rotation and translation, represented by the steep decrease of both delta values in the beginning and midsection of a subtask. On the other hand, both deltas decrease less steep in the final phase of each subtask. In combination with the significantly higher total translation, this fact indicates that participants overshooted the goal with the 3D mouse more often and therefore had to slow down to handle that.

Very fast simultaneous delta reduction in the beginning and midsection of a task with slower final adjustment phase

This can also be supported by the questionnaire results. Overall, participants rated it highest regarding speed. On the other hand, translation corrections were not that easy to perform than when using touch or the tangible in toggled mode where it was easier to just manipulate translation

Initially not easy to distinguish different input dimensions without rotation. Furthermore, the input method was not easy to understand initially. Participants commented that it took them some time getting used to controlling all input dimensions directly with one knob. Some participants stated that they found it difficult to only manipulate translation without rotation. Although the 3D mouse was the only relative input method tested, the coupling between input and output did not suffer from that. Overall, results of the 3D mouse highlight that designated hardware with direct mapping can improve the performance. Nevertheless, it can be difficult to precisely adjust a single input parameter using the knob. Additionally, the 3D mouse misses any customizability to adapt to specific virtual objects.

4.5.3 Tangible

Both tangible modes with significantly better performance than touch

Both tangible modes performed significantly faster than touch, so our first hypothesis H1 can be accepted. Furthermore, both modes led to significantly more rotation than using the 3D mouse or touch. This highlights, how easy it was to rotate the virtual object using a tangible. If for example colors of the cube had to be checked, the tangible could easily be turned around and back in a matter of seconds. In contrast to the other modes, both progress charts show local minima, which highlights that translation corrections sometimes enforced resetting the orientation. Regarding user ratings in general we found opposite opinions about both implemented modes. Overall, 9 participants rated one out of the two best. Splitted between both modes, participants usually liked on out of the two, rating the tangible condition best and the toggled condition worst or vice versa. This reflects personal preference as both modes on average performed similar. Depended on their developed tactics, some users may preferred the simplicity without any button, but others also found it tedious having to deal with the mirrored orientation at all times. For both modes users pointed out that they liked to get haptic feedback. Only by figures we consequently cannot put one mode clearly above the other one. Hereafter, properties related to the particular modes are discussed.

Continuous

The tangible input method was designed to connect input and output as close and simple as possible, with a continuous transmission of tangible orientation to virtual object rotation. Taking this drastic approach, we wanted to take a look at how it influences user tactics. As a result, the continuous mode was faster than touch, but slower than the 3D mouse. In general, the method polarized being ranked 4th place the most, but in contrast also being favored by 3 participants.

Closely couple tangible and virtual object by continuous mirroring of orientation

The progress chart 4.8 shows in a clear manner advantages and disadvantages of the mode. A frequently used tactic was to move the object to the target position first, indicated by the steep decrease of the translation. Once the position was close to the target indicated by a small position delta, the rotation was adjusted. If necessary, translation changes then led to a reset of the orientation. But once placed in position, the final rotation adjustment could be performed very fast, shown by the steep decrease of orientation delta towards the end. Consequently, this indicates the potential of the tangible, because large but very precise changes in rotation were possible in contrast to the adjustment phase of the 3D mouse. But on the other hand, any translation corrections were difficult to perform. Another characteristic to note is, that participants rotated the objects way more in this condition. Of course this is partly by design, since they were not able to stop the virtual object from rotating when they wanted to perform a translation. However, we still assume that the tangible mode encourages users to rotate the object more. In contrast, the total translation was significantly lower that in non-tangible modes. We suppose that absolute positioning can be more precise and participants took time to carefully position the object, as later corrections required a readjustment of rotation afterwards.

Object first moved to goal position and then rotated accordingly

Controlling the cube significantly faster than controlling the plane

The cube tangible was used to both control its virtual 3D model and an unrelated plane. When mirroring the tangible orientation directly to its controlled virtual object, we found out that participants performed significantly faster controlling the look-alike cube. In all other conditions, there was no such significant difference. Because of this data, H2 can be accepted. This shows that the strong association between input and output can help to position and rotate the object faster.

Translation corrections most difficult to perform

Ratings show that our third hypothesis H3 can be accepted as well. The difficulty of translation corrections was by far worst rated in this condition. On the other hand, the method was particular easy to understand with a close relation between input and output. The problem that every translation correction on-screen also resets the orientation is also supported by the questionnaire results and was commented frequently. All in all, this highlights that the tangible is intuitive to use in this mode, however the transition between on-surface and off-surface input is crucial for a fluent interaction.

Toggled

Decouple tangible and virtual object to ease translation corrections

By adding a toggle button we wanted to provide a mechanism to temporary decouple tangible and object orientation with the aim to ease this transition between on- and off-surface interaction. Regarding task completion times there was no difference to the continuous mode, but overall it received better user ratings being favored by 6 participants.

More freedom for users in developing their own tactics

The progress chart 4.9 shows, that delta translation and rotation develop different than in continuous mode. Effects of the continuous mode are dampened. As we also could observe during the study, some participants first tried to reach the correct orientation and then use the toggle to fix the orientation and move the object to the target. Others however moved it directly to the target and then tried to adjust the orientation, only using the toggle button if again translation corrections would be necessary. Some even moved the object next to the goal, adjusted the orientation, and finally dragged the object to the goal. This variety shows, that the toggle button allowed people to find their own fitting order of actions without enforcing one as the continuous mode did. As expected, total rotation was significantly lower than with the continuous mode, because the virtual object rotation could be fixed.

Translation corrections easier to perform with toggle

The results of the questionnaire indicate that the toggle mode could ease the problem of translation corrections. Additionally, 5 participants suggested to make corrections relative instead of absolute. This could ease small adjustments of already fixed and almost fitting orientations.

4.6 Design Recommendations

Design recommendations can be derived from study results

With the help of this study we could get first insights into tangible usage on- and off-surface. Quantitative and qualitative feedback have revealed general properties, as well as specific advantages and disadvantages of the implemented input methods. As a consequence, participants developed different tactics, which we could observe during usage and afterwards in the logged data. As a result, we can derive what already worked well and what could be improved when interacting with tangibles on-screen and in midair.

Give users a choice

First and foremost the comparison has shown, that users should be given a choice. Therefore, we would add a toggle button, but maybe not that prominently attached to the virtual object itself. Rather, it could be placed on the tangible itself. When ignored, the tangible could just be used the same way as in continuous mode.

Build custom tangibles looking like controlled object

Participants performed significantly faster when using the cube tangible to control its virtual counterpart. Therefore, we would recommend to customize the outer tangible shape to match its controlled object if possible. For example, a simplified 3D-model of the virtual object could be reused as blueprint for the 3D printer. It could guide participants by serving as a reference for their desired virtual manipulations. However, the plane object has shown that controlling an unequal-looking object is also possible.

Place pads on multiple tangible faces to minimize disturbance between on- and off-surface interaction

The transition between on-surface and off-surface interaction is crucial, because the screen works as a natural constraint forcing the tangible is a specific orientation when placed on it. In case this orientation is continuously transmitted, we want to keep the mental workload to recreate the previous orientation when lifting the tangible again as small as possible. Therefore, we strongly suggest to place pads on as many faces of the tangible as possible. In this way, the tangible becomes more flexible during usage as participants would not have to pay attention to which side must be facing the screen.

With toggle and non look-alike objects relative controls could be of advantage

If a tangible controls an arbitrary object and the orientation transmission can be toggled, it could be beneficial to process changes of orientation relative to the current state. This would avoid large jumps in orientation when deactivating the transmission, rotating the object and then activating the transmission again. In contrast, taking absolute values makes sense when controlling the look-alike object, as virtual and physical object should always match. For arbitrary objects like the plane in our study however, there usually is no indication where for example top and bottom are.

Independent translation and rotation control could benefit user performance

The performance of the 3D mouse has shown, that adjusting translation and rotation independent of each other and in parallel can significantly shorten the task completion time. With our tangible, translation on-screen could be performed by touch or with another transparent passive tangible. This would avoid occlusion and provide haptic feedback even when not focusing on the screen but on the midair interaction. On the other hand, changes like this can also be seen critical, as it would break the one-to-one association between tangible and virtual object and would occupy two-hands instead of one in the tangible continuous mode.

Chapter 5

Summary and Future Work

This final chapter concludes the thesis, summarizing our work on using tangibles on- and off-surface with the first use case of 3D object rotation. Furthermore, we sketch out future work as this new type of tangible interaction introduces a variety of potential research topics.

5.1 Summary and Contributions

We aimed at making tangibles usable on-surface and off-surface

Tangibles have already been used on-surface in many projects, providing input to virtual content when placed on a tabletop. With our work we now focused on additionally bringing tangibles off-surface. Putting this idea into practice, the task of 3D object rotation seemed like a good fit as a first sample application for combined on-screen and midair interaction. As chapter 2 has shown, tangibles have already been used for one-dimensional rotation input on-screen. For three-dimensional rotation control we collected various proposed approaches. Here, especially in immerse virtual environments tangibles have already been used as input method providing haptic feedback in otherwise virtual environments.

We extended hardware and software to use tangibles for 3D object rotation

For this thesis, we therefore constructed a tangible detecting its orientation in all three dimensions. Off-surface we used it to control object rotation, on-surface it was additionally capable to control object translation. In this way, we wanted to evaluate the crucial transition between both modes. For transmitting the data we tested two variants: one keeping virtual and physical object closely coupled applying orientation data continuously to the virtual object, the other one allowing to fix the virtual object orientation with a toggle button. To use them with our framework, we extended the software stack as described in section 3.1.

We implemented 3D mouse and touch methods to compare them to our tangible solution

Next, we implemented support for two alternative input methods as well. 3D mouse support (section 3.2) allowed us to compare our tangible solution to designated hardware used by 3D graphic designers. In addition to that, we designed a method using a virtual trackball and a rotation gizmo with 2DoF touch input (section 3.3) inspired by how 3D design tools handle 2DoF mouse input. Besides the general comparison of input methods, we wanted to analyze if a perceptual analogy between tangible and virtual object could have an impact on user performance. Therefore, we included two virtual objects to control: a cube looking like the tangible and a plane.

Our evaluation helped to detect characteristics of all input methods and to derive design recommendations

In our study, described in detail in chapter 4, participants had to perform tasks using all four input methods controlling both objects. We evaluated quantitative (task completion time, total rotation and translation, rotation and translation delta to goal over time, user questionnaire) and qualitative (user comments) feedback. Overall, the 3D mouse outperformed the other modes in terms of task completion time. Nonetheless, both tangible variants were significantly faster than using touch. Furthermore, they were easier to understand. The perceptual analogy between input and output led to a significantly increased performance when using the tangible with continuous orientation transmission. Concerning total translation, the tangible was significantly less translated than the 3D mouse. In contrast to that, the tangible was rotated significantly more than the other modes. Its translation and rotation progress showed that once the position on-screen was hit, orientation progressed tremendously fast to the goal. All this data was discussed as it highlights advantages (e.g. that tangibles could encourage rotation) and challenges (e.g. the forced orientation reset with input on-surface) regarding off-surface interaction. As a conclusion, we derived design recommendations. For example marker pads could be placed on all sides of the cube tangible to minimize the required rotation disturbance when translating on-screen. And that people should be given the chance to choose their own control method as the preference toggled versus non-toggled control depended on personal tactics.

Contribution of this thesis

This work contributes to the research on tangible user interfaces, making a first step towards combined on- and off-surface interaction on tabletops. Both proposed tangible input methods outperformed the 2DoF touch input for 3D object rotation. With the help of detailed data logging, we detected characteristics of different input methods. Subsequently, we could derive implications on the design of tangible user interfaces which should improve them in future. By that, a wide range of further research topics is opened as will be presented in future work.

5.2 Future Work

Control rotation and translation independently and in parallel inspired by 3D mouse performance

First, future work closest to the existing hard- and software state is presented. Based on the study results, it could be worth examining a two-handed control mode with object rotation and translation separated. Rotation would be controlled purely in midair while translation would be controlled independently on-screen with touch or a second tangible. This could be done without requiring new hardware or major modifications in the existing software.

Discrete mode selection associated with cube faces

We already suggested to put marker pads on all sides of the cube. This could also be used to implement a new input method using the different cube faces for discrete input. Each face could be associated with a mode or choice. Placing the tangible on-screen with a specific face down would then indicate a selection clearly visible on the tabletop. The cube would work as a physical constraint. Because only one face can be placed down at a time, an either-or decision is enforced. The laid down face could be detected actively with the integrated tracking sensors or passively with different marker patterns on each side.

Tangible awareness concerning midair interaction

This idea can also be connected to tangible awareness. It has already been shown that tangibles can increase awareness of each others actions on tabletops. Further research could examine the question if this effect is even strengthened by additional midair interaction.

Tangible on-surface for absolute control and off-surface for remote control to solve reachability issues on large tabletops

On large tabletops reachability is another challenge which could be approached by controlling virtual objects not only in an absolute manner on-surface, but also remotely if the tangible is lifted off-surface. As a remote, the tangible could then control the virtual object movement with its orientation. We could think of altering the roll axis for changing the direction, altering the pitch axis for changing the speed and altering the yaw axis for directly rotating the object around its z-axis.

Extend embedded hardware to enable new use cases

Up to this point, suggestions could be realized with the existing hardware. Next, future work requiring to extend the embedded hardware is laid out. First, we could think of improving the sensor accuracy with an additional magnetometer to counter the mentioned drift problem. Second, the embedded hardware could be extended to also track the tangible position in all three dimensions in the room and not just on the tabletop. Combing orientation and position data to provide 6DoF, the tangible could for example be used to control a camera for a virtual scene on the tabletop as shown in figure 5.1.

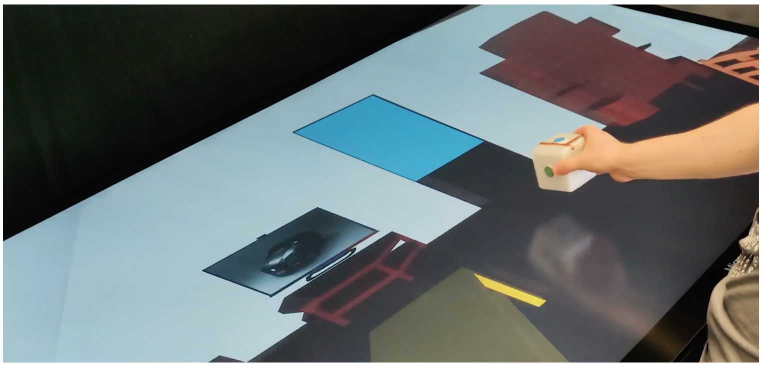


Figure 5.1: Tangible position in midair could be tracked to control both camera orientation and movement off-surface.

Transferring this approach to another demo task, the tangible could also control a spotlight which lights up content depended on its orientation and distance to the surface.

Slider control using tangible with different interpretations on- and off-screen

Camera or spotlight control are tasks focused on off-surface interaction, but of course other use cases could also include on-surface interaction. For example, a slider could be controlled by absolute tangible position on-screen and by relative tangible movement off-screen. The farer away from the surface the more precise the adjustments could be. This would reassemble common UI control schemes of today with tangibles.

Overall, this is just an excerpt of the most promising ideas. With all use cases the challenge remains how to fix midair input and how to combine it with on-screen interaction. All in all, bringing tangibles off-surface opens a wide range of new interaction schemes. By reaching a new dimension with tangibles, completely novel usage scenarios are possible.

Appendix A

User Study Consent Form and Questionnaire

The following consent form and questionnaire was handed out to participants during the user study. Before performing the tasks, users were asked to fill out the upper part. After performing all tasks on the tabletop, the second part including the ratings and the detail questions was filled out.

Informed Consent Form

Evaluating 3D object rotation using different input modes on a tabletop

PRINCIPAL INVESTIGATOR David Asselborn
Media Computing Group
RWTH Aachen University
david.asselborn@rwth-aachen.de

Purpose of the study: The goal of this study is to investigate user behaviour performing 3D object rotation and 2D object translation tasks on a tabletop using different input methods.

Procedure: Before the study, participants are asked to fill out a questionnaire with some information about themselves. Next, they will perform 8 tasks with 6 subtasks each, divided by short breaks. Before each new input mode there will be a short demo. After all tasks are finished, participants are asked to fill out a questionnaire about their experience. We will evaluate and store users interaction, timings, and results from the questionnaire.

Risks/Discomfort: You may become fatigued during the course of your participation in the study. After each task you have the possibility to rest and additional breaks between each target are also possible. There are no other risks associated with participation in the study. Should completion of either the task or the questionnaire become distressing to you, it will be terminated immediately.

Benefits: The results of this study will be useful for improving the user experience of tangible user interfaces.

Alternatives to Participation: Participation in this study is voluntary. You are free to withdraw or discontinue the participation.

Cost and Compensation: Participation in this study will involve no cost to you. There will be snacks and drinks for you during and after the participation.

Confidentiality: All information collected during the study period (questionnaire and data log related to each task) will be kept strictly confidential. You will be identified through identification numbers. No publications or reports from this project will include identifying information on any participant. If you agree to join this study, please sign your name below.

_____ I have read and understood the information on this form.
_____ I have had the information on this form explained to me.

Participant's Name	Participant's Signature	Date
	Principal Investigator	Date

If you have any questions regarding this study, please contact David Asselborn at email david.asselborn@rwth-aachen.de

Study 3D Object Rotation ID: __________

Gender: _______ Age: _______ Handedness: _______

Experience with 3D design?
◯ Yes ◯ Some ◯ No, if some/yes which tools: ____

Experience with 3D-Mouse?
◯ Yes ◯ Some ◯ No

Experience with tangible user interfaces?
◯ Yes ◯ Some ◯ No

Please rate the different input modes from 1 (= best) to 4 (= worst)

___ Touch
___ 3D-Mouse
___ Tangible with continuous rotation
___ Tangible with toggled rotation

Please answer questions about your experience with the different input modes

Touch

1. Translation corrections were easy to perform

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

2. Rotation corrections were easy to perform

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

3. The input method was easy to understand

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

4. The interaction was fast

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

5. The interaction was accurate

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

6. Input and output were closely coupled

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

Study 3D Object Rotation ID: _________

3D-Mouse

1. Translation corrections were easy to perform

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

2. Rotation corrections were easy to perform

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

3. The input method was easy to understand

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

4. The interaction was fast

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

5. The interaction was accurate

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

6. Input and output were closely coupled

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

Tangible with continuous rotation

1. Translation corrections were easy to perform

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

2. Rotation corrections were easy to perform

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

3. The input method was easy to understand

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

4. The interaction was fast

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

5. The interaction was accurate

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

6. Input and output were closely coupled

Totally disagree ◯ ◯ Neither nor ◯ ◯ Totally agree ◯

Study 3D Object Rotation ID: __________

Tangible with toggled rotation

1. Translation corrections were easy to perform

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

2. Rotation corrections were easy to perform

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

3. The input method was easy to understand

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

4. The interaction was fast

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

5. The interaction was accurate

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

6. Input and output were closely coupled

Totally disagree		Neither nor		Totally agree
◯	◯	◯	◯	◯

Comments?

Bibliography

Blender Foundation. Blender 2.79b. URL `https://www.blender.org`.

A. Cannavò, F. Cermelli, V. Chiaramida, G. Ciccone, F. Lamberti, P. Montuschi, and G. Paravati. T4t: Tangible interface for tuning 3D object manipulation tools. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 266–267, March 2017. doi: 10.1109/3DUI.2017.7893374.

Liang Chen and C. G. Brown. A 3D mouse for interacting with virtual objects. In *2005 IEEE International Symposium on Circuits and Systems*, pages 5011–5014 Vol. 5, May 2005. doi: 10.1109/ISCAS.2005.1465759.

Michael Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-D rotation using 2-D control devices. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pages 121–129, New York, NY, USA, 1988. ACM. ISBN 0-89791-275-6. doi: 10.1145/54852.378497. URL `http://doi.acm.org/10.1145/54852.378497`.

Christian Cherek, Anke Brocker, Simon Voelker, and Jan Borchers. Tangible awareness: How tangibles on tabletops influence awareness of each other's actions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 298:1–298:7, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3173872. URL `https://doi.org/10.1145/3173574.3173872`.

Matthias Ehlenz, Rene Roepke, and Ulrik Schroeder. Towards multi-touch learning applications in collaborative education. In *Proceedings of the 7th ACM International*

Symposium on Pervasive Displays, PerDis '18, pages 35:1–35:2, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5765-4. doi: 10.1145/3205873.3210709. URL `http://doi.acm.org/10.1145/3205873.3210709`.

Daniel Fallman, Anneli Mikaelsson, and Björn Yttergren. The design of a computer mouse providing three degrees of freedom. In Julie A. Jacko, editor, *Human-Computer Interaction. Interaction Platforms and Techniques*, pages 53–62, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73107-8.

George W. Fitzmaurice and William Buxton. An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 43–50, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258578. URL `http://doi.acm.org/10.1145/258549.258578`.

J. Franz, A. Menin, and L. Nedel. Lossless multitasking: Using 3D gestures embedded in mouse devices. In *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pages 109–116, June 2016. doi: 10.1109/SVR.2016.27.

H. Fukutake, Y. Akazawa, Y. Okada, and K. Niijima. 3D object layout by voice commands based on contact constraints. In *International Conference on Computer Graphics, Imaging and Visualization (CGIV'05)*, pages 403–408, July 2005. doi: 10.1109/CGIV.2005.2.

O. Ha and N. Fang. Development of interactive 3D tangible models as teaching aids to improve students' spatial ability in STEM education. In *2013 IEEE Frontiers in Education Conference (FIE)*, pages 1302–1304, Oct 2013. doi: 10.1109/FIE.2013.6685043.

Mark Hancock, Otmar Hilliges, Christopher Collins, Dominikus Baur, and Sheelagh Carpendale. Exploring tangible and direct touch interfaces for manipulating 2D and 3D information on a digital table. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 77–84, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.

1145/1731903.1731921. URL `http://doi.acm.org/10.1145/1731903.1731921`.

K. Henriksen, J. Sporring, and K. Hornbaek. Virtual trackballs revisited. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):206–216, March 2004. ISSN 1077-2626. doi: 10.1109/TVCG.2004.1260772.

Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. Usability analysis of 3D rotation techniques. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 1–10, New York, NY, USA, 1997. ACM. ISBN 0-89791-881-9. doi: 10.1145/263407.263408. URL `http://doi.acm.org/10.1145/263407.263408`.

Robert J. K. Jacob and Linda E. Sibert. The perceptual structure of multidimensional input device selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 211–218, New York, NY, USA, 1992. ACM. ISBN 0-89791-513-5. doi: 10.1145/142750.142792. URL `http://doi.acm.org/10.1145/142750.142792`.

Hyosun Kim, Georgia Albuquerque, Sven Havemann, and Dieter W. Fellner. Tangible 3D: Hand gesture interaction for immersive 3D modeling. In *Proceedings of the 11th Eurographics Conference on Virtual Environments*, EGVE'05, pages 191–199, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. ISBN 3-905673-21-5. doi: 10.2312/EGVE/IPT_EGVE2005/191-199. URL `http://dx.doi.org/10.2312/EGVE/IPT_EGVE2005/191-199`.

Myung-Soo Kim, Joon-Kyung Seong, Dae-Eun Hyun, Kang-Hoon Lee, and Yoo-Jin Choi. A physical 3D trackball. In *Proceedings Ninth Pacific Conference on Computer Graphics and Applications. Pacific Graphics 2001*, pages 134–138, 2001. doi: 10.1109/PCCGA.2001.962866.

T. P. Kurpjuhn, K. Nickels, A. Hauck, and S. Hutchinson. Development of a visual space-mouse. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 2527–2532 vol.4, 1999. doi: 10.1109/ROBOT.1999.773977.

Q. Lin, L. Yang, Y. Sun, T. Liu, X. Y. Li, and Y. Liu. Beyond one-dollar mouse: A battery-free device for 3D human-computer interaction via rfid tags. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1661–1669, April 2015. doi: 10.1109/INFOCOM.2015.7218546.

I. Scott MacKenzie, R. William Soukoreff, and Chris Pal. A two-ball mouse affords three degrees of freedom. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '97, pages 303–304, New York, NY, USA, 1997. ACM. ISBN 0-89791-926-2. doi: 10.1145/1120212.1120405. URL `http://doi.acm.org/10.1145/1120212.1120405`.

Márcio Martins, António Cunha, Irene Oliveira, and Leonel Morgado. Usability test of 3Dconnexion 3D mice versus keyboard + mouse in second life undertaken by people with motor disabilities due to medullary lesions. *Univers. Access Inf. Soc.*, pages 5–16, March 2015. ISSN 1615-5289. doi: 10.1007/s10209-013-0329-9. URL `http://dx.doi.org/10.1007/s10209-013-0329-9`.

Thi Thuong Huyen Nguyen and T. Duval. Poster: 3-point++: A new technique for 3D manipulation of virtual objects. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 165–166, March 2013. doi: 10.1109/3DUI.2013.6550230.

S. Reifinger, F. Laquai, and G. Rigoll. Translation and rotation of virtual objects in augmented reality: A comparison of interaction devices. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 2448–2453, Oct 2008. doi: 10.1109/ICSMC.2008.4811662.

A. M. Rodrigues, O. Belloc, E. Z. Borba, M. Nagamura, and M. K. Zuffo. Batmen beyond: Natural 3D manipulation with the batwand. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 258–259, March 2017. doi: 10.1109/3DUI.2017.7893370.

I. Seidinger and J. Grubert. 3D character customization for non-professional users in handheld augmented reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 129–134, Sept 2016. doi: 10.1109/ISMAR-Adjunct.2016.0059.

Ken Shoemake. ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface '92*, pages 151–156, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 0-9695338-1-0. URL `http://dl.acm.org/citation.cfm?id=155294.155312`.

Philip Tuddenham, David Kirk, and Shahram Izadi. Graspables revisited: Multi-touch vs. tangible input for tabletop displays in acquisition and manipulation tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2223–2232, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753662. URL `http://doi.acm.org/10.1145/1753326.1753662`.

Simon Voelker, Kosuke Nakajima, Christian Thoresen, Yuichi Itoh, Kjell Ivar Overgård, and Jan Borchers. PUCs: Detecting transparent, passive untouched capacitive widgets on unmodified multi-touch displays. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 101–104, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512791. URL `http://doi.acm.org/10.1145/2512349.2512791`.

Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Overgård, and Jan Borchers. PERCs: Persistently trackable tangibles on capacitive multi-touch displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 351–356, New York, NY, USA, 2015a. ACM. ISBN 978-1-4503-3779-3. doi: 10.1145/2807442.2807466. URL `http://doi.acm.org/10.1145/2807442.2807466`.

Simon Voelker, Kjell Ivar Øvergård, Chat Wacharamanotham, and Jan Borchers. Knobology revisited: A comparison of user performance between tangible and virtual rotary knobs. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, ITS '15, pages 35–38, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3899-8. doi: 10.1145/2817721.2817725. URL `http://doi.acm.org/10.1145/2817721.2817725`.

Malte Weiss, Julie Wagner, Roger Jennings, Yvonne Jansen, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. Slapbook: Tangible widgets on multi-touch tables in groupware environments. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 297–300, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-493-5. doi: 10.1145/1517664.1517725. URL `http://doi.acm.org/10.1145/1517664.1517725`.

D. J. Zielinski, D. Nankivil, and R. Kopper. Specimen Box: A tangible interaction technique for world-fixed virtual reality displays. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 50–58, March 2017. doi: 10.1109/3DUI.2017.7893317.

Index

Typeset September 28, 2018