ARCube: Enabling Passive Haptic Feedback for Augmented Reality User Interfaces Using a Cube-Shaped Controller

Bachelor thesis by Jonas Julian Wombacher
Date of submission: April 20, 2022

1. Review: Prof. Dr. Max Mühlhäuser
2. Review: Dominik Schön
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Computer Science Department
Telecooperation Lab
HCI

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Jonas Julian Wombacher, die vorliegende Bachelorarbeit gemäß §22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 20. April 2022

J. Wombacher

Zusammenfassung

Augmented Reality (AR) verbindet die virtuelle mit der physischen Welt und ermöglicht es, rein virtuelle dreidimensionale Objekte als Teil der physischen Welt darzustellen. Dennoch bleibt die virtuelle Umgebung digital und ist somit für die Nutzer nicht greifbar. Um mit der digitalen Welt zu interagieren, müssen Nutzer aktuell mit ihren Händen Gesten in der Luft ausführen. Während dieser Ansatz zwar vielseitig und für verschiedene AR-Anwendungen anpassbar ist, können virtuelle Benutzeroberflächen den Nutzern aber kein haptisches Feedback bieten. Derzeit erfahren die Nutzer nur visuelle und auditive Hinweise, wenn sie mit der Benutzeroberfläche interagieren. Das Fehlen von physischem Feedback kann zu niedrigerer Präzision führen, da die Nutzer nichts fühlen können, wenn sie mit digitalen Objekten interagieren. Das kann bei der Bedienung von AR-Benutzeroberflächen besonders kritisch sein. Um haptische Wahrnehmung bei der Bedienung von digitalen AR-Benutzeroberflächen zu ermöglichen, stellen wir ARCube vor, einen würfelförmigen Controller für die Steuerung von AR-Benutzeroberflächen. Diese Thesis befasst sich mit der Gestaltung und Beurteilung eines würfelförmigen Controllers für AR-Interaktion. Der ARCube ist mit verschiedenen Arten von physischen Steuerelementen, wie Tastern, Scrollrädern, Touchpads und weiteren ausgestattet. Das ermöglicht den Nutzern, passives haptisches Feedback zu erfahren, während sie mit AR-Benutzeroberflächen interagieren. Im Rahmen dieser Thesis haben wir mögliche Steuerelemente klassifiziert und eine Beispielanwendung entwickelt, um das Konzept eines solchen Eingabegeräts in einer Nutzerstudie mit 14 Teilnehmern zu bewerten. Die Studie ergab, dass ein derartiger Controller die Performance des Nutzers verbessern und, in Kombination mit Eye-Tracking, die wahrgenommene Arbeitsbelastung verringern kann.

Abstract

Augmented reality (AR) bridges the gap from the virtual world to the physical world, allowing purely virtual 3D objects to appear as part of the physical world. However, the virtual environment remains digital and thus not tangible for the users. Interacting with these digital worlds requires users nowadays to perform midair hand gestures. While this approach is very versatile and customizable for the respective AR applications, users do not experience any haptic feedback from the virtual user interface. Currently, the users only receive visual and auditive cues when performing inputs. The absence of physical feedback can lead to lower precision of the input, as users can't feel when they interact with the digital bits. This can be especially critical for inputs to AR user interfaces. To provide haptic sensation for interacting with digital AR user interfaces, we introduce ARCube, a cube-shaped controller for input in AR user interfaces. This thesis covers the creation and evaluation of a cubic controller usable for AR Input. The ARCube is fitted with different kinds of physical input controls, like buttons, scroll-wheels, touch-enabled surfaces, etc. This allows users to experience passive haptic feedback, while interacting with AR user interfaces. As part of the thesis, we classified possible input controls and developed a sample application to evaluate the concept of such an input device in a user study with 14 participants. In this study, we found that such a controller can improve the user's performance and, when paired with eye tracking, reduce the perceived workload.

Contents

1 Introduction **7**

2 Related Work **9**
2.1 Gorilla Arm Syndrome . . . 9
2.2 Heisenberg effect . . . 10
2.3 Passive-Haptic Feedback . . . 11
2.4 Cube-shaped controllers . . . 12
2.5 Other approaches at extended reality input . . . 14

3 Methodology **16**
3.1 Controller . . . 16
3.2 Application . . . 17
3.2.1 Baseline without controller . . . 18
3.2.2 Controller with hand tracking . . . 18
3.2.3 Controller with eye tracking . . . 19

4 Implementation **21**
4.1 Controller . . . 21
4.2 Communication between Controller and Application . . . 24
4.3 Application . . . 24
4.3.1 Baseline without controller . . . 26
4.3.2 Controller with hand tracking . . . 26
4.3.3 Controller with eye tracking . . . 28

5 Study **29**
5.1 Participants . . . 29
5.2 Design . . . 29
5.3 Measures . . . 31
5.4 Procedure . . . 32

6 Results 33
6.1 Quantitative Evaluation 34
6.2 Task Load Index 39
6.3 Qualitative Feedback 40

7 Discussion 43
7.1 Performance 43
7.2 Perceived workload and personal preference 45

8 Summary and future work 47
8.1 Limitations 47
8.2 Future Work 48

1 Introduction

Augmented Reality (AR) has a broad spectrum of research and application areas. To name a few examples, there is research in the medical field, like applying AR in the „*rehabilitation of post-stroke patients*“[11] or controlling surgical robots[38]. Education can benefit from AR as well, for instance when teaching the basics of English to children with a different mother tongue[12] or by visualizing information in order to motivate children, when they are learning[24]. In the context of marketing, AR can for example provide more individual advertisements in shopping malls[16] or help with building a brand [13].

Currently, gestures are a common way of interacting with the virtual world of AR[2]. Wearable AR systems, like AR smart glasses[31], allow users to experience the combination of the virtual and physical world without being bound to stationary monitors or having to hold smartphones in front of them. When paired with wearable AR systems, like Microsoft's HoloLens 2[1] with built-in hand tracking capabilities, gesture control can offer a versatile and unobtrusive way to interact with virtual objects, not requiring additional hardware in the form of controllers. All you have to do is put on the AR glasses and start the desired application.

The drawback of gesture input, however, is that it can be slower, less accurate and more fatiguing, compared to other input concepts, utilizing for example mouse and keyboard[8] or dedicated controllers[5, 22]. Because of this, there have been various approaches on how to improve gesture input, among them identifying less fatiguing gestures[20] and adding additional sensors to allow for more precise and less fatiguing gestures[14].

While such approaches can reduce fatigue and improve accuracy to some degree, they still work without physical controllers and are therefore unable to provide haptic sensations to the users. This can be circumvented by taking advantage of the environment's haptics[10, 40], for example by letting the user drag his finger across a wall instead of moving it through the air. Doing so does not require physical controllers, but it depends on the

[1]https://www.microsoft.com/en-us/hololens/hardware, last visited: 03.04.2022

environment providing appropriate surfaces, which can limit the locations suitable for using the system. Physical controllers, on the contrary, are not bound to the environment, which means they are able to provide haptic feedback anywhere.

Whether not needing to carry a controller or experiencing haptic sensations is more important, naturally depends on the intended use case. If, for example, the user constantly needs to be able to use both of his hands to work on something, alternating between picking up a controller and putting it aside all the time would be very disruptive. If, however, most of the activities happen in the virtual world with prolonged menu interactions, a physical, haptic controller might be the better choice, potentially getting rid of the fatiguing nature of midair gestures.
Apart from that, input concepts with haptic feedback can also be faster and more accurate than concepts without it[1, 26, 5]. Based on such observations, we designed and built two lightweight, cube-shaped controllers featuring the following physical input modalities: a button, a momentary rocker-switch, a rotary encoder, a touchpad and a trackball.

In this thesis, we present how we created those controllers (Figure 1.1), as well as how we tested them on AR user interfaces in a user study.
The second chapter contains related work, succeeded by the third chapter with the methodology of the controllers and the accompanying application. The fourth chapter provides details on the implementation of the controllers and the application. Chapters five and six explain the user study we conducted and its results, respectively. The results are discussed in chapter seven. Chapter eight concludes this thesis with a summary of our approach and future work to be done.

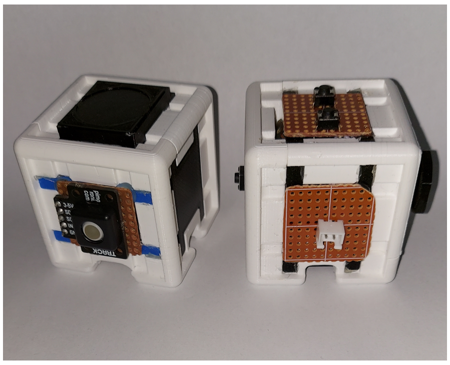


Figure 1.1: The two controllers

2 Related Work

The following four sections are going to outline the foundations of the key considerations for our approach: Gorilla Arm Syndrome, Heisenberg effect, Passive-Haptic Feedback and Cube-shaped controllers. After that, we are going to present some examples, of how other researchers approached user input for extended reality, also taking into account one or more of these considerations.

2.1 Gorilla Arm Syndrome

Hand gestures are a common way to interact with objects and menus in augmented realities. The augmented reality system tracks the user's hands and uses this information to recognize a set of predefined gestures. These gestures are then mapped to actions in the virtual world, for example, performing a grabbing motion and then moving the hand to grab a slider's handle and change its value by moving the handle.
Such gestures typically require the user to reach out in front of his body. Doing this over and over for an extended period of time causes the so-called „*Gorilla Arm Syndrome*“[17]. Hansberger et al. describe the symptoms as „*arm fatigue and a feeling of heaviness in the arms*“[17]. They compared three different approaches to controlling a video game with regard to the amount of exertion. The approaches were keyboard input, standard gestures performed in midair and special gestures, which were executed while resting the arms on a chair's armrest. The experiment showed, that such special gestures are a significant improvement over typical in-air gestures, causing amounts of exertion similar to using a keyboard.
However, the special gestures used in this experiment can not be universally applied in augmented reality systems, as the experiment used a standard monitor for displaying the game and a separate motion capture system using inertial measurement units. A common and lightweight technique for hand tracking in augmented reality uses cameras embedded

in the front of the head-mounted display, for example in the Microsoft HoloLens 2. This requires the user's hands to be positioned inside the camera's field of view, which typically means they have to be extended in front of the body. Gestures performed while resting the arms close to the body would not be recognizable in such a setup.

2.2 Heisenberg effect

Another challenge for input systems in augmented reality can arise, when a one-handed input device is used. It was named „*Heisenberg effect*" by Bowman et al.[4], who described it as „*the phenomenon that on a tracked device, a discrete input (e.g. button press) will often disturb the position of the tracker. For example, a user wants to select an object using ray casting. She orients the ray so that it intersects the object, but when she presses the button, the force of the button press displaces the ray so that the object is not selected*"[4].
They prevented this effect by using a two-handed input system. The users wore two gloves, which meant that one hand could be used to choose a target, while the other hand was used to trigger the action without affecting the first hand's orientation.

Wolf et al.[39] conducted further research about the Heisenberg effect. Based on their experiment, through which they evaluated the impact of the Heisenberg effect under different circumstances, they presented some more approaches to reduce its impact, including the following:

a) Increasing the size of the selection targets could reduce the number of wrong selections, because after this a pointer displacement with the same value is less likely to actually cause the user to miss his desired target.

b) Instead of executing the selection with the latest pointer position available, the position from the point in time, where the button press just started, could be used. In the conducted experiment, however, this approach only improved error rates in the stationary pointing tests. In the ballistic pointing tests, it created more errors than the Heisenberg effect did.

c) After observing the offsets caused by the Heisenberg effect, this data could be utilized „*to create correction vectors that can be subtracted from the selection position*"[39], countering the unwanted displacements and thus reducing the error rate.

2.3 Passive-Haptic Feedback

Lindeman et al.[26] aimed to improve the way users can interact with 2D user interfaces in virtual 3D spaces. Their idea was to use „*Passive-Haptic Feedback*"[26], which they described as follows: „*Passive-haptic "devices" are physical objects which provide feedback to the user simply by their shape, texture, or other inherent properties. In contrast to active haptic feedback systems, the feedback provided by passive-haptic feedback devices is not controlled by a computer. These objects can be either rigid or deformable*"[26].
The physical object they chose was a paddle. The user could tap it and slide his finger across it, as opposed to pointing and moving his finger in the air. In the user study, the participants had to perform actions typically required for user interfaces, clicking buttons and „*Drag-and-Drop*"[26] gestures. Those actions had to be executed with and without Passive-Haptic Feedback in the form of the paddle.
The user study resulted in faster, as well as more accurate interaction while the paddle was used, confirming the effectiveness of Passive-Haptic Feedback for 2D user interfaces in virtual 3D spaces.

Besançon et al.[1] compared three input techniques for manipulating 3D objects, using mouse and keyboard, a touchscreen and tangible input with a cuboctahedron. They conducted a user study, which required the users to manipulate a 3D object in order to reach the given position and orientation.
The techniques were evaluated with regard to the following categories:

a) The „*Task Completion Time*"[1] confirmed, that the tangible input was significantly faster than both other techniques.

b) „*Accuracy*"[1] did not reveal significant differences between any of the techniques.

c) The „*Fatigue*"[1] was rather similar in all techniques as well.

d) According to the „*Workload*"[1], tangible input is less demanding than using a touchscreen. The results of tangible compared to mouse and keyboard and mouse and keyboard compared to touchscreen were too close to draw conclusions.

e) As for „*Preferences*"[1], the tangible technique was the favorite one of most of the users.

f) „*The Impact of Experience*"[1] showed, that compared to users without experience, experienced ones achieved a better „*Accuracy*"[1] with all three techniques. „*Task Completion Time*"[1] was significantly shorter for mouse and keyboard only.

Besançon et al. concluded with several advantages and disadvantages for each of the techniques, deeming all three suitable for 3D object manipulation. While they were equally accurate, tangible input had the benefit of being the fastest input technique.

Passive haptic feedback is not only beneficial for generic 2D user interfaces or 3D object manipulation. Shaer et al.[33] described common concrete application areas for Tangible User Interfaces.
One of them is the application in learning. Among other reasons, this is because addressing more senses at once can improve the learning effects of children. More specifically targeted at teaching programming to children, tangibles can help with understanding abstract concepts needed for programming.
Another area is problem-solving and planning, where tangibles can „*facilitate mental work*", „*communicate interaction syntax*" and „*limit the solution space*"[33].
Besides those areas, Shaer et al. also explained the application in information visualization, entertainment, play and edutainment, music, social communication and tangible reminders and tags.

2.4 Cube-shaped controllers

Once you decide to use tangibles in a project, the question arises, which physical shape the tangibles should have. Lefeuvre et al.[25] examined the popularity of cubic shapes for tangible devices. During their research, they recognized three particularly important papers for this topic: „*Exploring Cube Affordance: Towards a Classification of non-verbal Dynamics of Physical Interfaces for Wearable Computing*"[34], „*Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*"[21] and „*The i-Cube: Design Considerations for Block-based Digital Manipulatives and their Applications*"[15].
Lefeuvre et al. identified nine advantageous properties of cubes: „*Manipulation as Input, Placement in Space as Input, Arrangement, Multifunctionality, Randomness, Togetherness & Variations, Physical Qualities, Container, and Pedestal for Output*"[25].

- When it comes to *Manipulation as Input*, the cube, with its identical faces being connected through right angles, can be easily mapped to and manipulated in the three-dimensional space.
- *Multifunctionality* refers to the identical faces of a cube, allowing to scatter different functionalities across it.

- Following this Multifunctionality, a cube can introduce *Randomness* into a Tangible User Interface. By handling the cube like a die, one of the functionalities placed on its faces can be randomly selected.
- Once multiple cubes are in use, their placement and orientation relative to each other can be interpreted as user input. As an example of this property, the *Placement in Space as Input*, Lefeuvre et al. show a combination of three cubes, representing different inputs, depending on which of the cube's faces are directed towards the other cubes.
- Also relying on the presence of multiple cubes, *Arrangement* enables to combine them to build bigger structures. By adjusting, which specific cubes are part of the structure and in which orientation they are arranged, the input behavior mapped to the structure can be modified.
- *Togetherness and Variations* regards more than a single cube as well. Instead of to the faces of a single cube, different functionalities can also be assigned to different cubes. Sharing the same practical cubic shape, these tangibles can easily be applied alternately to perform various tasks.
- While the previous properties all considered the functionality and application of tangibles, the *Physical Qualities* take a tangible's fabrication into account. The composition of identical, flat faces joined by right angles allows for an easy assembly utilizing sheets of various materials, for example wood.

Other than these „*formal properties, resulting directly from the geometrical characteristics*“[25], the following final two aspects were described as „*symbolic and semantic properties that exist due to the meaning humans relate to cubic shapes*“[25].

- The first association is a cube as a *Container*, enclosing non-tangible content in its tangible shape. Utilizing this, separate cubes could therefore be assigned different information, for example in order to let them represent images or textual data.
- The second concept is the *Pedestal for Output*. Cubes can house diverse output devices like speakers or vibrators, while hiding their individual appearances. This allows users to solely concentrate on the produced output instead.

The rest of this section will be used to showcase a few examples, of how some of these cube properties can be observed in research about cube-shaped controllers.
Block et al.[3] built a cubic TV remote, taking advantage of the Manipulation as Input and Multifunctionality properties. The cube's faces represented different TV channels,

which the user could choose by rotating the controller.
Working with the same two cube properties, Terrenghi et al.[36] developed a cube-shaped controller for multiple-choice quizzes. The controller featured a display on every face, one of which was used to show the question, while the remaining ones displayed possible answers. A chosen answer could be submitted by shaking the controller.
Making use of Manipulation as Input and Multifunctionality as well, Roudaut et al.[32] presented their Rubikon, a Rubik's cube with embedded rotary encoders. Its possible applications included navigating user interfaces and manipulating 3D objects by rotating the different parts of the cube.
The same two properties can also be seen in the CubeLendar project by Matviienko et al.[28]. They designed and built a cube-shaped calendar, which can notify the user about appointments through LED signals. The cube was populated with LEDs, seven-segment displays and a matrix display. Its faces were assigned various kinds of information, so that one face, for example, showed the weather forecast for the time of the appointment, while another one showed details about the type of appointment. Controlling the calendar worked by rotating it, in order to cycle through days or appointments.
Utilizing multiple cubes, therefore being able to apply both the Togetherness and Variations and the Arrangement properties, Camarata et al.[9] created a system for tourists to navigate digital galleries. This system allowed users to trigger the display of different information by applying different cubes. Combining two cubes also combined the information associated with both cubes.

2.5 Other approaches at extended reality input

Kharlamov et al.[23] utilized smartwatches as controllers for a virtual reality game. The user could point at objects with his arm wearing the smartwatch, which was then translated into a pointer in the game. This was achieved with ray casting based on data from the smartwatch's inertial measurement unit.
First, actions could be triggered by tapping on the smartwatch's touchscreen. However, this induced the Heisenberg effect. In the final version, different actions could be performed by rotating the forearm clockwise or counterclockwise by at least 40°. In order to prevent the Heisenberg effect in this version, the pointer's position was locked, once a forearm rotation of 20° was recognized, possibly indicating an ongoing full 40° rotation. By doing this, the pointer displacement caused by rotating the forearm was reduced to an amount small enough for the use case in the virtual reality game.
Their take on decreasing the Gorilla Arm Syndrome was, that „*users are encouraged to*

interact by moving the forearm, keeping the elbow close to the body, and, if needed, reposition their entire body, instead of making wide movements with their shoulder joint"[23]. However, it was only an assumption, that this significantly improves on the Gorilla Arm Syndrome, which they did not investigate within the scope of this paper.

With a focus on object manipulation in AR, Bozgeyikli et al.[5] compared hand gestures, the Magic Leap 1[1] controller and a tangible cube with each other. The cube was built by housing the same Magic Leap 1 controller in a 3D-printed enclosure.
In the conducted study, the participants had to interact with a virtual cube, which had the same dimensions as the tangible cube. The tasks they had to perform consisted of moving and rotating the virtual cube, in order to match the wanted position and orientation.
The results of the study showed, that the tasks could be completed significantly faster with the controller or the tangible cube than with hand gestures. Comparing the speed of the controller and the tangible cube did not result in significant differences.

Potts et al.[29] created a toolkit for building a tangible cubic controller, which, among other platforms, can be used for AR interaction. User input could be given through touch gestures on the controller's surfaces. These gestures were recognized utilizing „*Capacitive Sensing*"[29]. As the recognizable gestures were predetermined through the specific construction of the 3D-printed faces, the controller featured interchangeable faces, allowing to customize it for different applications.
Potts et al. further presented their toolkit by explaining, how their controller can be applied to some exemplary use cases: „*a media controller, a platform game, and a 3D model inspection tool*"[29].

[1]https://www.magicleap.com/en-us/magic-leap-1, last visited: 03.04.2022

3 Methodology

3.1 Controller

Our controller was given a cubic shape. This decision was mainly based on two of the properties identified by Lefeuvre et al.: *Manipulation as Input* and *Multifunctionality*[25]. The equally sized faces of a cube make it easy to distribute various combinations of input modalities across them. The cube's 90° angles mean that it is simple as well as unambiguous to switch between the different modalities. In addition, both in development and later on in usage, it is clearly distinguishable, which face is looking in a certain direction. Having equal sides without one primary, bigger side, allows a cube to be held in any orientation, without forcing a preferred way to use it. This does not only allow the user to arrange the input modalities how he likes, but it also makes it possible to manipulate how a particular modality functions. A rocker-switch, for example, which provides left- or right-input in the cube's neutral orientation, could therefore give up- or down-input instead while the cube is rotated by 90°.
As an added benefit, the *Physical Qualities* property[25] helped with prototyping the controller. Being able to build a cube out of perfboards and therefore simultaneously wiring the electronic circuits directly on the inside of the cube's shell, meant that no separate boards for easy wiring were required. Precise and space-saving wiring was only needed, once the controller's design was finalized.

The input modalities to be placed on the controller were chosen to cover different categories of Buxton's taxonomy[7].
Both Button and Momentary Rocker-Switch are one-dimensional, mechanical devices working with the position property. The Rotary Encoder is also one-dimensional and mechanical, but represents the motion property.
Touchpad and Trackball are two-dimensional devices, where the Touchpad senses position through touch, and the Trackball senses motion mechanically.

The classification of these input modalities with Buxton's taxonomy is visualized in figure 3.1.

		Number of Dimensions						
		1			2		3	
Property Sensed	Position	Button	Momentary Rocker-Switch					Mechanical
					Touchpad			Touch
	Motion			Rotary Encoder		Trackball		Mechanical
								Touch
	Pressure							
								Sensing Type

Figure 3.1: Input modalities classified with Buxton's taxonomy[7]

3.2 Application

Augmented Reality user interfaces are often constructed similar to 2D desktop user interfaces, so we selected a set of traditional user interface elements known from 2D desktop applications. This set contained buttons, checkboxes, drop-down menus, rotary knobs, on-off toggles, radio buttons, sliders and color selection dialogues. The content, which the application was later filled with for the study, was created using these eight types of elements.
In this thesis, three main input concepts are being compared:

1. Baseline without controller: This concept works with the users' hands only, representing the typical way of controlling augmented reality through hand gestures.
2. Controller with hand tracking: The second concept includes a controller while still using hand tracking. The baseline's gestures are replaced with controller input, but hand tracking is kept for determining the controller's position.
3. Controller with eye tracking: The final concept combines controller input with eye tracking, completely discarding hand tracking.

3.2.1 Baseline without controller

Because this concept works without a controller, all interaction happens through hand gestures. The user is supposed to be able to interact with all elements, except for sliders and rotary knobs, with his fingers by performing a midair pushing motion at the position, where the element is displayed to him. A slider should be controlled by reaching out to its handle, performing a pinching motion in order to grab the handle, moving the hand along the slider's axis to change its value and reversing the pinching motion to let go of the handle. For rotary knobs, the procedure is similar, but instead of moving the hand along an axis, the hand needs to be rotated in order to turn the knob accordingly.

Characteristics regarding Gorilla Arm Syndrome, Heisenberg effect and Passive-Haptic Feedback

a) Gorilla Arm Syndrome: As the user has to constantly perform gestures reaching out in front of his body when interacting with the user interface, we expect the Gorilla Arm Syndrome to have a negative impact here.

b) Heisenberg effect: Because no physical input modalities are used here, we do not expect unwanted selections caused by pointer displacements, which, for example, could occur when pressing a physical button.

c) Passive-Haptic Feedback: Since this would require a physical object of some form, the baseline will not provide Passive-Haptic Feedback.

3.2.2 Controller with hand tracking

For the second concept, we replaced hand gestures with controller input, but otherwise kept the hand tracking. This concept is designed for the user to select elements of the user interface by moving his hand, holding the controller, next to it. While his hand is next to an element, the user can interact with it using the controller and its input modalities.
More specifically, clicking or tapping on the physical button, the trackball or the touchpad should trigger buttons, choose radio buttons, toggle on-off toggles, (un-)check checkboxes and open and close drop-down menus as well as color selection dialogues.
The values of sliders and rotary knobs should be adjustable through the rotary encoder.
The momentary rocker-switch should enable jumping between radio buttons or checkboxes and moving the selection within opened drop-down menus or color selection dialogues.

Moving one's finger across the touchpad or rotating the trackball should combine the interaction possibilities of both the rotary encoder and the momentary rocker-switch.

In addition to the physical input modalities, it should be able to turn rotary knobs by rotating the whole controller itself.

Characteristics regarding Gorilla Arm Syndrome, Heisenberg effect and Passive-Haptic Feedback

a) Gorilla Arm Syndrome: We suppose, that this is going to have a negative impact here as well, because the user has to reach out to the different elements of the user interface.

b) Heisenberg effect: The controller contains physical input modalities, while being operated by the same hand selecting the target. We assume, that this could lead to unwanted interactions.

c) Passive-Haptic Feedback: Being a physical object with physical input modalities, the controller will provide Passive-Haptic Feedback.

3.2.3 Controller with eye tracking

The final concept relies on the same controller input, as the previous one. However, hand tracking is not used here at all.
Eye tracking has proven itself to be a promising way of user input for various augmented and virtual reality applications[27, 37, 35, 30]. Most importantly for our use case, it allows user input without relying on hand movement, which could be beneficial when it comes to the Gorilla Arm Syndrome and the Heisenberg effect.
Therefore, this input concept utilizes eye tracking to allow the user to select elements by looking at them. As long as an element is being looked at, the user should be able to interact with it using the controller.
The physical input modalities should fundamentally trigger the same actions here as they did in combination with hand tracking. Yet, reaching out and rotating the whole controller should not turn rotary knobs anymore, as this was linked to the hand tracking.

Characteristics regarding Gorilla Arm Syndrome, Heisenberg effect and Passive-Haptic Feedback

a) Gorilla Arm Syndrome: Contrary to the previous two approaches, this one relies on eye tracking instead of hand tracking. This means, the user is able to control the user interface while keeping his arms in the position most comfortable for him, which should greatly reduce the Gorilla Arm Syndrome.

b) Heisenberg effect: We expect, that separating the hand holding the controller from the element selection is going to prevent the Heisenberg effect, as the user's eyes are not affected by interacting with physical input modalities like buttons.

c) Passive-Haptic Feedback: Just like in the previous concept, the physical controller with its physical input modalities will provide Passive-Haptic Feedback.

4 Implementation

4.1 Controller

According to the decision about the shape explained in the previous chapter, the controller was planned and constructed with a cubic shape. A 3D printed cube frame with a 50 mm outer side length was used as the shell of the controller. As a cable for data transmission and power supply would have hindered its free rotation, the controller was designed for wireless operation.
The space inside the hollow cube frame was used to house the microcontroller. For the microcontroller, an Arduino Nano 33 BLE[1] was chosen, because of its built-in wireless capabilities and inertial measurement unit (IMU) as well as its compact form factor.
For populating the cube's faces, the input modalities illustrated in figure 3.1 were implemented as follows.
For the button, a simple push button was connected to the microcontroller. The momentary rocker-switch was created by placing two of the same kind of push buttons next to each other.
Common basic rotary encoders are often over two or three centimeters high, which means that the encoder would be sticking out of the side of the controller significantly, considering the controller's shell itself is only five centimeters wide. Therefore, a rotary mouse wheel encoder, laid flat onto the controller, with a custom 3D printed wheel was used instead. This combination of encoder and wheel only stuck out of the controller's side by about one centimeter.
A small trackball was ordered from Pimoroni[2]. For the touchpad, a circular trackpad with a 23 mm diameter[3] was chosen. This touchpad was used with a one layer thick 3D printed cover, which improved its accuracy.

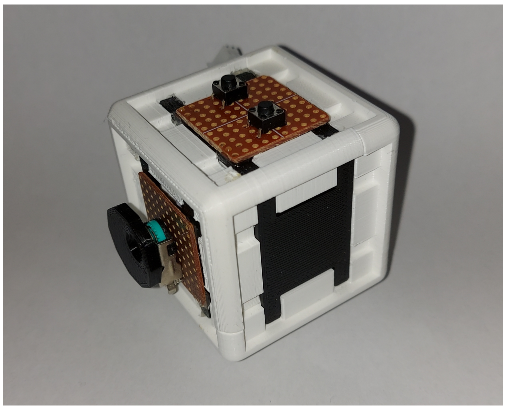
[1]https://docs.arduino.cc/hardware/nano-33-ble, last visited: 03.04.2022
[2]https://shop.pimoroni.com/products/trackball-breakout, last visited: 03.04.2022
[3]https://www.cirque.com/glidepoint-circle-trackpads, last visited: 03.04.2022

On the software side, the SensorFusion library[4] was used. This allowed to obtain accurate pitch, roll and yaw values by combining the inertial measurement unit's accelerometer, gyroscope and magnetometer. These values, along with the data from the connected input modalities, were transmitted to the application with Bluetooth Low Energy (BLE).

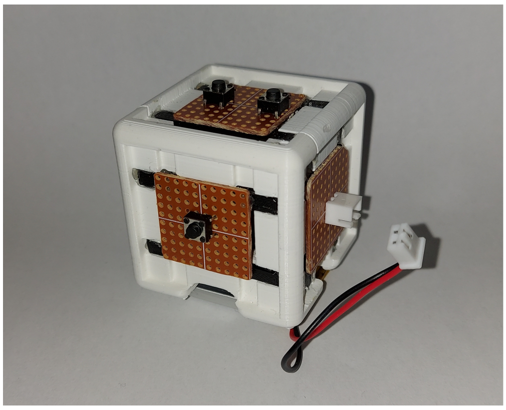
During the construction of the controller, it turned out, that the space inside the cube frame would not have been enough to wire all five physical input modalities to the micro-controller. Increasing the controller's size would have made it harder to operate it with one hand. Instead, a second controller was built and the components were distributed among the two controllers. Thus, the first one received the single push button, the momentary rocker-switch and the rotary encoder, while the second one got the remaining trackball and touchpad.

Figure 4.1 shows the two controllers. In figure 4.1a you can see the first controller, with the momentary rocker-switch on top and the rotary mouse wheel encoder on the controller's side facing left. The black surface on the right side is just a cover. Figure 4.1b shows the first controller as well, this time from another point of view. The momentary rocker-switch is sitting on the top again, while the side facing left in this picture holds the single push button. This face is the opposing face of the one with the rotary encoder. The right side houses the battery connector, allowing to quickly turn the controller on or off. The second controller, as seen in figure 4.1c, has the touchpad on top and the trackball on its side, facing left in the image. Both controllers are placed next to each other in figure 4.1d.

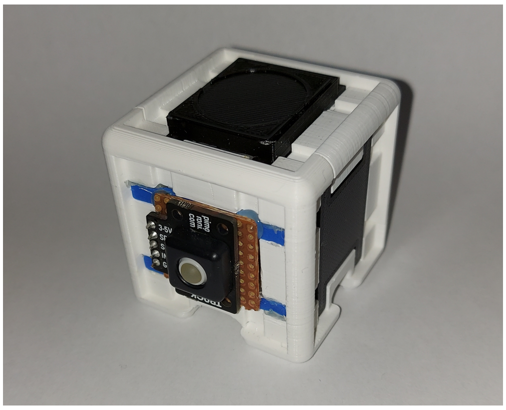
[4] https://github.com/aster94/SensorFusion, last visited: 03.04.2022



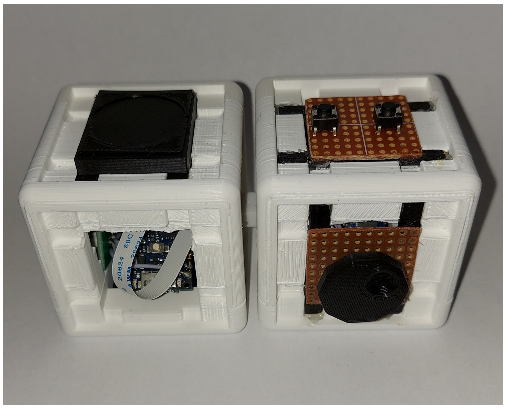
(a) The first controller. Visible here: momentary rocker-switch (on the top) and rotary encoder (on the left side)



(b) The first controller. Visible here: momentary rocker-switch (on the top), single button (on the left-facing side) and battery connector (on the right-facing side)



(c) The second controller. Visible here: trackball (on the side) and touchpad (on the top)



(d) Both controllers next to each other

Figure 4.1: The two controllers

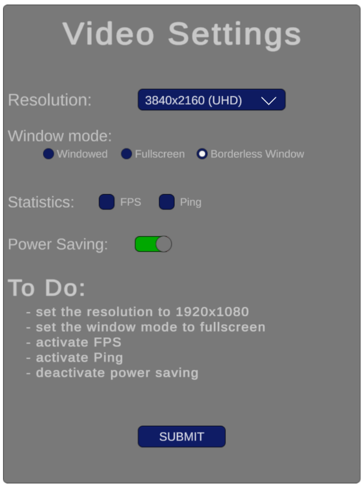
4.2 Communication between Controller and Application

The data transfer from controller to application was organized in three BLE characteristics, which acted like channels for different kinds of data.
The first characteristic was used for the single button, the momentary rocker-switch, the rotary encoder and the trackball. It was updated, whenever a user input with one of the associated modalities occurred. This data was then directly used by the application, treated like input-events triggering interaction with the user interface.
The second characteristic was exclusively assigned to the touchpad. It was continuously updated with the current finger position on the touchpad. Similarly, the last characteristic was permanently updated to contain the current pitch, roll and yaw values produced by the inertial measurement unit.
Other than with the first characteristic, the data from these two characteristics went through further computations on the application site. These computations are going to be explained in section 4.3.2.

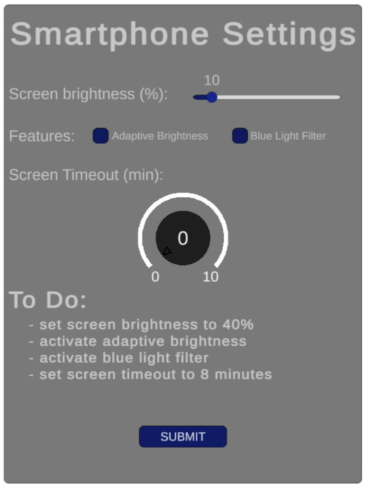
4.3 Application

The application was developed for and run on a Microsoft HoloLens 2. It was built using Unity version 2020.3.14f1 and Microsoft MixedRealityToolkit (MRTK) version 2.7.2 [5].
The content of the application consisted of settings tabs, which contained the user interface elements from the set introduced earlier: buttons, checkboxes, drop-down menus, rotary knobs, on-off toggles, radio buttons, sliders and color selection dialogues. The goal of the application was to interact with these settings tabs one after another.
Figure 4.2 contains sample settings tabs, showcasing the different user interface elements. Color selection dialogues can be found separately, in figure 4.3. Here you can see the two different modes of color selections, which are going to be explained in section 4.3.2 as well.

[5] https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/v2.7.2, last visited: 03.04.2022

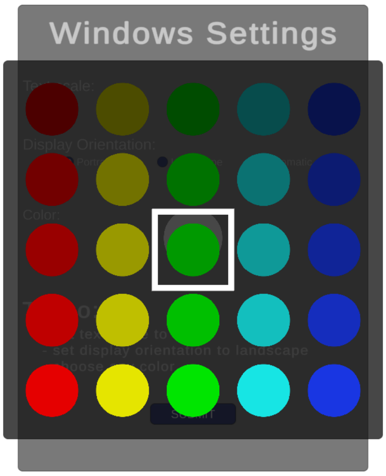


(a) Settings tab with a drop-down menu, radio buttons, checkboxes, an on-off toggle and a button (top to bottom)

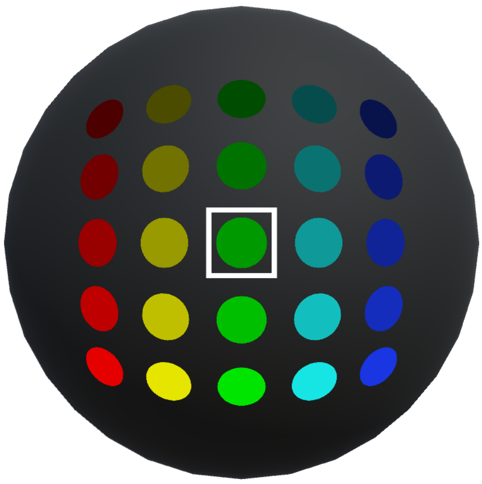


(b) Settings tab with a slider, checkboxes, a rotary knob and a button (top to bottom)

Figure 4.2: Examples of settings tabs



(a) Opened color selection dialogue, showing the plane with colors



(b) Opened color selection dialogue, showing the sphere with colors

Figure 4.3: The two possible appearances of opened color selection dialogues

The input concepts required different implementations of the individual user interface elements:

4.3.1 Baseline without controller

As the baseline was supposed to represent the typical way of augmented reality interaction using hand gestures, this concept relied on user interface elements provided by the MRTK. Buttons, checkboxes, on-off toggles, radio buttons and sliders were directly available in the MRTK.
Drop-down menus were created out of a button, alternating between displaying and hiding a group of radio buttons.
The color selection dialogues were built similar to the drop-down menus, however the button displayed and hid a group of buttons in the form of clickable color images. The arrangement of these color images is shown in figure 4.3a.
Finally, the rotary knobs utilized grabbable and manipulable objects included in the MRTK, reading their orientation to obtain the knobs' values.

4.3.2 Controller with hand tracking

In contrast to the baseline, the two input concepts relying on a controller did not require direct interaction between the user's hands and the individual user interface elements. Instead, the elements were operated through the data received from the controller, for example triggering a button in the user interface when the physical button on the controller was pressed.
Because of this, the standard user interface elements provided by Unity itself could be used. Those included buttons, checkboxes, drop-down menus, radio buttons and sliders. On-off toggles as well as rotary knobs were not available and had to be implemented from scratch, utilizing images and manipulating their position and rotation to display the element's current state.
The color selection dialogues used a button, that switched between displaying and hiding a set of color images along with an image of a frame. The frame acted as an indicator for the current selection, being moved to surround the according color image, each time another color was chosen. This set of color images can be seen in figure 4.3a. As an alternative to these color images, which all shared a common plane, a color selection sphere was implemented, visible in figure 4.3b. It could be displayed or hidden by the button as well. The sphere also contained color images, but they were laid on top of its

surface. In order to select a color, the sphere could be rotated to bring the desired color into the center of the user's field of view. This alternative was created for more intuitive control when using the physical trackball. Thus, color selection dialogues showed the sphere, when they were activated by the trackball, and the basic color images, when any physical input modality but the trackball was used.
The functionality for the hand tracking was provided by the MRTK. Whenever the hand position was needed, the MRTK was queried for the current position of the tip of the user's left or right index finger, depending on which was visible to the HoloLens' cameras.
Hand tracking was applied for two things. First, user interface elements were selected through the position of the user's hand. Whenever data from the controller was received, the element closest to the hand was the one to be controlled with the received data. Second, some user interface elements relied on the hand position when they were interacted with:

- A group of radio buttons could be clicked to choose the radio button closest to the hand.
- A group of checkboxes could be clicked to toggle the value of the checkbox closest to the hand.
- An opened color selection dialogue could be clicked to choose the color closest to the hand. This was only supported by the basic color image dialogue and not by the sphere.
- A rotary knob could be controlled by rotating the controller, if the hand was close to the knob.

The controller's orientation was derived from the transmitted pitch, roll and yaw values. This orientation was further processed for two purposes.
First, the data from directional physical input modalities, including the rotary encoder, the momentary rocker-switch, the touchpad and the trackball, was adjusted to the orientation. This made sure, that the triggered interactions were in line with the directions expected by the user, no matter how the controller was being held. For example, if the user swiped upwards on the touchpad in his perception, an upward interaction should be triggered. This should always be the case, even if the touchpad was upside down and it therefore technically registered a downward swipe.
Second, a rotary knob could be controlled by rotating the controller. This interaction required, that the user started pressing down the single button or the trackball earlier while holding the controller close to the knob. As long as the button or trackball continued to be pressed down, the controller's rotation was transferred onto the knob. Releasing the button or trackball also decoupled the knob from the controller's rotation. This procedure

aimed to simulate physically grabbing, rotating and then releasing a rotary knob.
The data received from the touchpad was processed in different ways, depending on the context:

- For opening or closing drop-down menus or color selection dialogues, toggling on-off toggles or checkboxes and clicking buttons, a click was triggered, when the user touched the touchpad and let go of it, without significantly moving his finger in the meantime.
- In order to change the selection among radio buttons or multiple checkboxes, within opened drop-down menus or within opened color selection dialogues, the offsets between finger positions were calculated. When the user's finger for example moved right since the last registered position, one step to the right was triggered.
- If a slider or rotary knob was to be controlled, the horizontal position of the user's finger relative to the horizontal borders of the touchpad was directly translated into a concrete value for the slider or knob. This way, the user could slide his finger left and right as if he was actually moving a slider.

4.3.3 Controller with eye tracking

This concept is similar to the previous one, where a controller and hand tracking were used. The controllers as well as the user interface elements were the same. However, hand tracking was replaced with eye tracking.
Eye tracking was available in the MRTK as well. This made it possible to keep track of which object was being looked at currently, through an event triggering every time the user started looking at a new object. This information could then be used to interact with the correct user interface element, when input data from the controller was received.
Without the hand tracking, grabbing and rotating a knob by rotating the controller, as well as choosing the element closest to the hand, like with the radio buttons, was not available in this input concept. Everything, that did not use hand tracking, however, for example adjusting the directions of the physical input modalities to the orientation of the controller, was still the same here.

5 Study

5.1 Participants

14 participants took part in the study, with seven of them being female and seven being male. The participants' ages ranged from 18 to 56 years, with a mean of 26.7 and a standard deviation of 11.6. At the beginning of the study, each participant filled in a form. Besides age and gender, the form asked for previous experience with Augmented Reality and for previous experience with other controllers, for example Xbox or Wii controllers. Figure 5.1 shows the distribution of those prior experiences.

	Frequencies				
	no experience	slight experience	intermediate experience	regular user	proficient user
Experience with AR	9	5	0	0	0
Experience with other controllers	0	2	6	3	3

Figure 5.1: Distribution of the participants' prior experiences with AR and other controllers

5.2 Design

The study included one independent variable: the input method. The three options for this independent variable were the methods explained earlier: baseline (BL), controller with hand tracking (HT) and controller with eye tracking (ET).
In order to balance out carryover effects, the orders, in which the participants tested the

input methods, followed a balanced Latin square (BLS). The BLS was calculated with a generator [1] based on Bradley's paper[6].

For the study, the application was filled with six settings tabs per input method, for a total of 18 tabs. Because each participant was supposed to test both controllers, the tabs were split into two sets of nine tabs. Both of those sets contained three tabs per input method each. Every tab contained some UI elements, as well as a To-Do list. The To-Do lists explained to the user, into which state he should bring the UI elements, for example to which value a certain slider should be set.
The different types of UI elements, that were placed on the settings tabs, were distributed evenly across the three input methods. For example, on all tabs for the baseline combined, there were a total of six buttons. Correspondingly, there were six buttons on all tabs for the hand tracking method combined, as well as on those for the eye tracking method combined. The complete distribution of UI elements can be seen in Figure 5.2.

			Input Methods			
			Baseline BL	Hand tracking HT	Eye tracking ET	Total
UI Elements	Sliders	20 Steps	2	2	2	6
		5 Steps	1	1	1	3
	Rotary Knobs	20 Steps	1	1	1	3
		5 Steps	1	1	1	3
	Radio Buttons	4 Options	2	2	2	6
		3 Options	3	3	3	9
	Drop-down Menus	6 Options	1	1	1	3
		5 Options	1	1	1	3
		4 Options	2	2	2	6
	On-Off Toggles		6	6	6	18
	Checkboxes	4 Boxes	2	2	2	6
		2 Boxes	1	1	1	3
	Color Selection Dialogues		2	2	2	6
	Buttons		6	6	6	18
	Total Elements		31	31	31	93

Figure 5.2: Distribution of UI elements on the settings tabs for all input methods

While the settings tabs were being interacted with, the application wrote entries into a log file. A new entry was logged each time one of the following events occurred:

- A new settings tab was loaded.
- A user interface element was focused by either looking at it or holding the hand close to it.

[1] https://cs.uwaterloo.ca/~dmasson/tools/latin_square/, last visited: 03.04.2022

- An element was interacted with, for example a button was pressed or a slider's value was changed.

Each log entry contained the following information about the event:

- The current participant's ID.
- The timestamp of when the event occurred.
- The name of the object that triggered the event.
- The type of event.
- Potentially the new value as well as the desired value of the object. For example, that a slider was set to position 10 and that the participant is supposed to set it to 25 eventually.

5.3 Measures

From the data logged during the study, we calculated the following measures:

- *Interaction Duration:* This measure represents the averaged time, in seconds, that it took the participant to interact with one single UI element.
- *Correctness:* This value is the fraction of the UI elements in the study, that the participant managed to finally set to the expected target value.
- *Overstepping:* The averaged number of times, per UI element, that the participant stepped over the expected target value, i.e. he moved the element away from the already correct value. For a slider with a target value of 4, for example, setting its value to 3, then 4 and then 5 would mean overstepping one time.

Apart from that, we applied the NASA Task Load Index (TLX) [19] in the study, in order to evaluate the subjective workload perceived by the participants.

5.4 Procedure

All participants started with filling in the form about age, gender and prior experiences. Then the controllers, as well as the study's procedure, were explained to them.
Following these explanations, the participants put on the HoloLens 2 and performed the built-in eye calibration, adjusting the eye tracking to their eyes. Once the eye calibration was completed, the participants continued with the study's tutorial, which taught them how to interact with the different UI elements using the controllers.
After that, they were shown one settings tab after another, grouped by the tabs' input methods. The BLS mentioned in section 5.2 was used to balance out carryover effects between the input methods. Therefore, the order of input methods, which the participants used while interacting with the settings tabs, was determined by the BLS. A participant with the BLS-sequence *BL-HT-ET* for example, had to interact with three baseline tabs first, followed by three hand tracking tabs and finally three eye tracking tabs. In this case, the next participant would have been assigned the sequence *BL-ET-HT*.
During the first nine tabs, one of the controllers was connected. Following changing to the other controller, the sequence of input methods was then repeated with new settings tabs. Each time the input method changed, the participants filled in a NASA TLX form[2], resulting in six forms per participant. The controller, which was used first, alternated with every participant.
After completing these two parts of the study, the participants were asked for qualitative feedback, based on three guiding questions. First, they were asked to rank the three input methods, from best to worst. Then, they were asked to name at least one positive aspect of their whole experience in the study, that they especially liked, and at least one negative aspect, that particularly bothered them. Finally, they decided, which of the two controllers they liked more.

[2]https://humansystems.arc.nasa.gov/groups/TLX/, last visited: 03.04.2022

6 Results

For the quantitative evaluation of the data collected during the study, we considered the different types of UI elements as separate tasks. This allowed a more specific evaluation of the input methods' impacts on UI interaction.

For checkboxes, drop-down menus, on-off toggles, radio buttons and color selection dialogues we created one task each, for which we examined the correctness and overstepping measures. Interaction duration was not considered here, as it would have been difficult to extract the uniform starting and ending points of the interactions needed for a fair comparison between the input methods.

Sliders and rotary knobs both appeared in the study with different fidelities, either with a total of five (low) or twenty (high) possible values per slider or knob. Therefore, we created two tasks for sliders and rotary knobs each, grouped by their fidelity. For these four tasks, we investigated correctness, overstepping and interaction duration. Interaction duration could be compared here, because sliders and rotary knobs have consistent step-by-step interactions between grabbing and releasing them.

Buttons were not explicitly compared here, as they only trigger actions, in the case of this study advancing to the next tab, instead of having a target value they should be set to. Additionally, the kind of interaction with buttons was essentially the same as with on-off toggles.

For the results of the NASA TLX forms, we did not split the types of UI elements into separate tasks. Instead, we directly compared the results between the three input methods, as the participants filled in a form, whenever the input method was changed.

The figures used in the following two sections are boxplots visualizing the median as the horizontal line within the box, the 25th percentile as the lower border of the box and the 75th percentile as the upper border of the box. The ends of the vertical lines above and below the box show the maximum and minimum values excluding outliers, while the dots above or below the box represent potential outliers[1].

[1]https://www.r-graph-gallery.com/boxplot.html, last visited: 03.04.2022

6.1 Quantitative Evaluation

For all of the following tasks, we performed a repeated measures ANOVA for non-parametric data using jamovi[2]. The post-hoc comparisons, also executed in jamovi, were based on Durbin-Conover.

Checkboxes

For the correctness of checkboxes, there was no significant difference between BL, HT and ET ($\chi^2 = 0.286$, $df = 2$, $p > 0.05$). The oversteps did not show significant differences either ($\chi^2 = 3.00$, $df = 2$, $p > 0.05$).

Drop-down menus

The tests for the drop-down menus did not result in significant differences, neither for correctness ($\chi^2 = 4.00$, $df = 2$, $p > 0.05$), nor for oversteps ($\chi^2 = 2.00$, $df = 2$, $p > 0.05$).

On-off toggles

The correctness of on-off toggles showed significant differences between the input methods ($\chi^2 = 12.4$, $df = 2$, $p < 0.01$). According to the post-hoc tests, there was a significant difference between BL and HT ($p < 0.01$) and between HT and ET ($p < 0.001$), but not between BL and ET ($p > 0.05$). Figure 6.1 visualizes the correctness for the different input methods, with the correctness in HT being significantly lower than in BL and ET. BL shows a wider spread of values than ET, but this was not enough to cause significant differences.

As there were zero oversteps of on-off toggles for all three input methods, no significant differences could be detected here.

[2]https://www.jamovi.org/, last visited: 03.04.2022

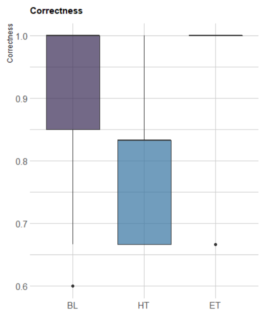


Figure 6.1: Correctness of on-off toggles (higher is better)

Radio buttons

We found no significant differences for the correctness of radio buttons ($\chi^2 = 0.500$, $df = 2$, $p > 0.05$).

The oversteps, however, showed significant differences ($\chi^2 = 18.1$, $df = 2$, $p < 0.001$). The post-hoc tests resulted in significant differences between BL and HT ($p < 0.001$), between BL and ET ($p < 0.001$) and between HT and ET ($p < 0.01$). These differences can be seen in figure 6.2. BL caused significantly less oversteps than HT, while HT in turn still caused significantly less than ET.

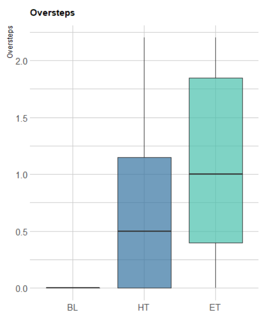


Figure 6.2: Oversteps of radio buttons (lower is better)

Color selection dialogues

The color selection dialogues did not show significant differences, both for correctness ($\chi^2 = 6.00$, $df = 2$, $p = 0.050$) and for oversteps ($\chi^2 = 2.00$, $df = 2$, $p > 0.05$).

Sliders with low fidelity

Sliders with only five possible values exhibited no significant differences, neither for interaction duration ($\chi^2 = 0.571$, $df = 2$, $p > 0.05$), nor the correctness ($\chi^2 = 2.00$, $df = 2$, $p > 0.05$), nor the oversteps ($\chi^2 = 5.07$, $df = 2$, $p > 0.05$).

Sliders with high fidelity

For sliders with twenty possible values, on the other hand, we did find significant differences for the interaction duration ($\chi^2 = 9.57$, $df = 2$, $p < 0.01$) and the oversteps ($\chi^2 = 17.0$, $df = 2$, $p < 0.001$).

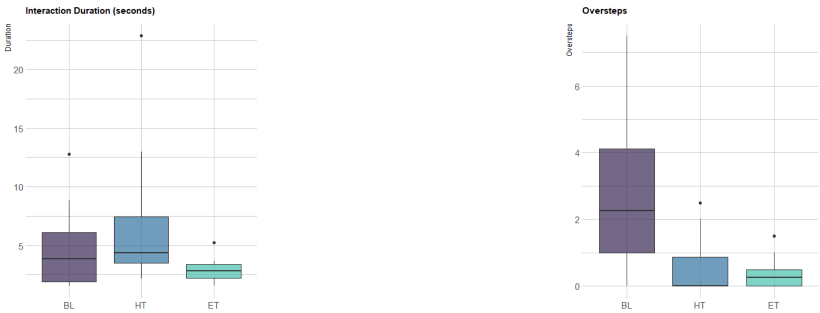
The post-hoc tests for interaction duration showed significant differences between BL and HT ($p < 0.05$) and between HT and ET ($p < 0.01$), but not between BL and ET ($p > 0.05$). Figure 6.3a shows BL and ET having similarly low interaction durations, while HT has significantly higher interaction durations.

According to the post-hoc tests for oversteps, there were significant differences between BL and HT ($p < 0.001$) and between BL and ET ($p < 0.001$), but not between HT and ET ($p > 0.05$). These differences are visualized in figure 6.3b, where HT and ET both show significantly less oversteps than BL.

All high fidelity sliders were finally set to the correct target value, which means no significant differences for the correctness could be identified.

Rotary Knobs with low fidelity

For rotary knobs with only five possible values, we found significant differences for the interaction duration ($\chi^2 = 15.4$, $df = 2$, $p < 0.001$). In the post-hoc tests, there were significant differences between BL and HT ($p < 0.001$) and between BL and ET ($p < 0.001$),



(a) Interaction durations of high fidelity sliders (b) Oversteps of high fidelity sliders

Figure 6.3: Measures for high fidelity sliders (lower is better)

but not between HT and ET ($p > 0.05$). The interaction durations can be seen in figure 6.4. HT and ET both required low interaction durations, while BL required significantly higher interaction durations.

Neither the correctness ($\chi^2 = 2.00$, $df = 2$, $p > 0.05$), nor the oversteps ($\chi^2 = 4.88$, $df = 2$, $p > 0.05$) showed significant differences here.

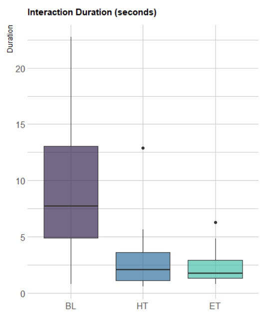


Figure 6.4: Interaction durations of low fidelity rotary knobs (lower is better)

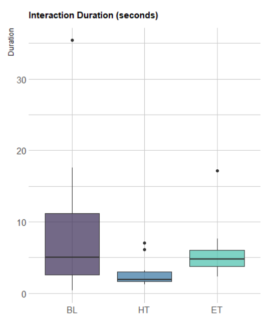
Rotary Knobs with high fidelity

For rotary knobs with twenty possible values, we observed significant differences for the interaction duration ($\chi^2 = 10.1$, $df = 2$, $p < 0.01$) and the oversteps ($\chi^2 = 13.9$, $df = 2$, $p < 0.001$).

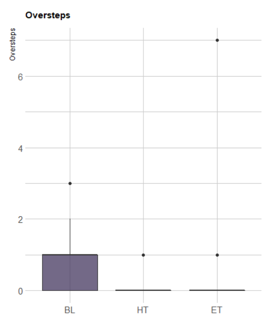
In the post-hoc tests for interaction duration, we saw significant differences between BL and HT ($p < 0.01$) and between HT and ET ($p < 0.01$), but not between BL and ET ($p > 0.05$). Figure 6.5a contains these differences, with HT taking significantly less interaction duration than BL and ET.

The post-hoc tests for oversteps resulted in significant differences between BL and HT ($p < 0.001$) and between BL and ET ($p < 0.001$), but not between HT and ET ($p > 0.05$). This is visualized in figure 6.5b, where HT and ET show similarly few oversteps, as compared to the significantly more oversteps in BL.

Because all high fidelity knobs finally had the correct value, no significant differences in the correctness could be detected.



(a) Interaction durations of high fidelity rotary knobs



(b) Oversteps of high fidelity rotary knobs

Figure 6.5: Measures for high fidelity rotary knobs (lower is better)

6.2 Task Load Index

We evaluated the results of the NASA TLX forms in two ways.
First, we calculated a „*Raw TLX*“[18] by adding the values of the six categories included in a NASA TLX. This resulted in three values per participant, one for each of the input methods. These combined values allowed us to perform a quick overall comparison of the subjective workloads for the different input methods.
The repeated measures ANOVA for non-parametric data revealed significant differences in the RAW TLX ($\chi^2 = 11.4$, $df = 2$, $p < 0.01$). The Durbin-Conover post-hoc tests showed significant differences between ET and BL ($p < 0.05$) and between ET and HT ($p < 0.001$), but not between BL and HT ($p > 0.05$). Figure 6.6 contains the boxplot for the Raw TLX, with ET having a significantly lower workload than BL and HT.

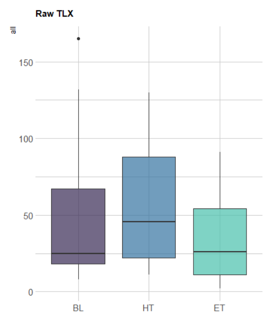
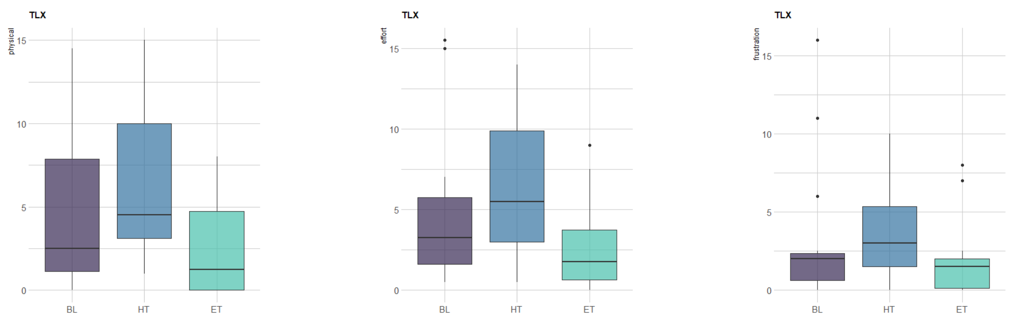


Figure 6.6: Raw TLX (lower is better)

Second, we compared the input methods in separate tests for each of the six categories. This was done in order to provide a more specific comparison, investigating, which aspects of the perceived workload exhibited significant differences between the methods, and which did not.
Conducting repeated measures ANOVA for non-parametric data showed no significant differences for the mental demand, temporal demand and performance ($p > 0.05$).
For physical demand ($\chi^2 = 14.8$, $df = 2$, $p < 0.001$), effort ($\chi^2 = 11.5$, $df = 2$, $p < 0.01$) and frustration level ($\chi^2 = 10.7$, $df = 2$, $p < 0.01$), however, we did find significant differences. In line with the Raw TLX, the post-hoc tests for these three categories all resulted in significant differences between ET and BL and between ET and HT, but not between BL and HT:

- Physical demand: $p < 0.01$ for ET-BL, $p < 0.001$ for ET-HT and $p > 0.05$ for HT-BL
- Effort: $p < 0.05$ for ET-BL, $p < 0.001$ for ET-HT and $p > 0.05$ for HT-BL
- Frustration level: $p < 0.05$ for ET-BL, $p < 0.001$ for ET-HT and $p > 0.05$ for HT-BL

These results are visualized in figure 6.7. ET has a significantly lower value than BL and HT for all three measures. While the figure shows slightly better values for BL than for HT, these differences were not significant.



(a) TLX: physical demand (b) TLX: effort (c) TLX: frustration level

Figure 6.7: TLX: comparisons of physical demand, effort and frustration level (lower is better)

6.3 Qualitative Feedback

Using the controller in connection with eye tracking (ET) was the most popular input method among the participants, as eleven out of the fourteen participants ranked it first. The decision between second and third place for the baseline (BL) and the controller with hand tracking (HT) was often a tight decision. Ultimately, the baseline was a bit more popular than HT. The whole distribution of the participants' input method ranking is visualized in figure 6.8.
One negative impression of the baseline was the higher effort, both overall and, more specifically, for high fidelity rotary knobs, as stated by participant 4: „*I could not rotate the knobs far enough in one go, so I had to grab and rotate them multiple times to reach the target*"[P4]. Another problem with the baseline was, that the gestures performed by the

users were not recognized sometimes, for example with participant 5: „*Sometimes, I had to grab the sliders multiple times, because it was not recognized*“[P5].
When using the controller with hand tracking, some participants felt like having to hold the controller out in front of them was a bit more annoying compared to reaching out with just their hands in the baseline.

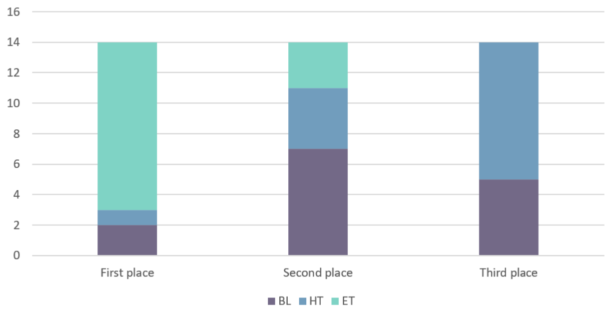


Figure 6.8: Ranking of the input methods by the participants

When asked, which of the two controllers they liked more, ten participants chose the first one, with the physical buttons and the rotary encoder, while only two chose the second one, with trackball and trackpad. The remaining two participants could not decide between the controllers, as they found both approaches equally good.
As reported by the participants, this clear decision was mainly due to the fact, that the input modalities on the first controller provided direct, physical feedback. The buttons clicked, both audible and tangible, whenever they registered a press, and the rotary encoder moved in distinct, tangible steps when turned. Participant 2 for example said, that „*it clicks, and you immediately know, that exactly one input was registered*“[P2].
Trackpad and trackball, on the other hand, do not provide this kind of feedback, except for the trackball clicking when pressed. Therefore, the interaction can feel imprecise, when you do not know exactly, when and how many inputs are triggered when sliding your finger across the trackpad for example. This was also stated by the participants: „*Using the touchpad felt imprecise and confusing*“[P6].
However, there also were two participants, who explicitly liked the trackball, because of its versatility by combining directional and clicking inputs. Participant 11 for example stated, that „*the trackball is very sensitive, but once you adapt to that, it works great*“[P11].

As for the study itself, about interacting with user interfaces in AR, we received positive feedback from the participants. „*It was more fun than interacting with menus on a* *laptop, even if it has a touchscreen*“[P14]. For many of them, it was their first experience with AR. They were excited by the virtual user interface being integrated into the real world, allowing them to physically move closer to it for example, instead of everything being fixed on the screen, moving with them if they rotate their head or move.

7 Discussion

7.1 Performance

Our study showed, that adding passive haptic feedback through physical input modalities on a controller does not outperform standard hand gestures in all cases. But, while there are cases, where the baseline worked better than the controllers, especially for the HT method, the controllers mostly matched the baseline or improved on it. Out of the tested user interface elements, sliders and rotary knobs seem to be the best use cases for the controllers.

Checkboxes, drop-down menus, on-off toggles, radio buttons and color selection dialogues

When interacting with hand gestures, user interface elements like checkboxes, drop-down menus, on-off toggles, radio buttons and color selection dialogues all rely on tapping gestures, just like buttons do. As long as the elements you want to interact with are reasonably large, this is a rather simple gesture, as you only have to hit the targeted object once, without any precise follow-up motions. This could be an explanation for the fact, that the results of our study largely show no significant differences between the three input methods for those kinds of UI elements. The only two of those elements, that did show significant differences, were on-off toggles and radio buttons.
For on-off toggles, the correctness was similarly high with BL and ET, but a bit lower in HT. This could be due to the Heisenberg effect we expected for HT. A scenario for errors caused by the Heisenberg effect, in this case, could be, that the user is about to interact with an element next to an on-off toggle. As he presses a button on the controller, this press causes his hand with the controller to move a bit, selecting the adjacent on-off toggle instead of the intended element and changing the on-off toggle's value without noticing

it. This does not pose a problem with BL, because such unintended movements do not occur without physical input modalities like the buttons. It does not occur with ET either, because the element selection happens through eye tracking, decoupled from potential unintended hand movements.
For the radio buttons, there were significant differences regarding the oversteps. While there were no oversteps in BL, there were some in HT and ET. In BL, radio buttons are controlled by individual tapping gestures, without a particular risk of accidentally changing the value once the correct value is chosen. By adding sensitive physical input modalities like the touchpad and the trackball, HT and ET added the risk of triggering more inputs than intended, resulting in more oversteps. Input modalities like the rotary encoder and buttons probably do not have this problem. The scale of our study was, however, not large enough to provide enough data to compare the input modalities in this regard.

Sliders and rotary knobs

UI elements like sliders and rotary knobs are controlled by more complicated hand gestures than just tapping. The user needs to grab the handle, keep moving or rotating his hand, until the targeted value is reached, and then release the handle again. For such elements, adding passive haptic feedback in the form of physical input modalities on a controller is an improvement. This was confirmed by the results of our study.

For low fidelity versions of these elements, the results largely showed no differences. This makes sense, as the large steps between different values do not require as precise hand gestures. Still, the interaction duration for low fidelity rotary knobs in BL was higher than in HT and ET. The reason for this could be, that it takes more effort to reach out, grab and rotate your whole hand, than it takes to rotate a small rotary encoder for example. This difference in interaction duration was not observed for the low fidelity sliders, which could be, because moving your hand a bit to the left or right takes less effort than rotating your hand while holding it in the same place. Therefore, there is not as much effort that can be saved by using the controller as there is for the rotary knobs.

The high fidelity versions showed more differences. Both for sliders and rotary knobs, there were more oversteps in BL than in HT and ET, with a bigger difference for the sliders than the knobs. This is in line with the passive haptic feedback of the input modalities allowing for a more precise and predictable interaction. For the gesture input, the small steps of the high fidelity elements pose a bigger challenge, as more precise gestures are required, which ultimately leads to accidentally moving or rotating the slider or knob too

far.
When it comes to the interaction duration, the values for HT were significantly worse than for BL and ET with the high fidelity sliders. But, with the high fidelity rotary knobs, the values for HT were significantly better than for BL and ET. We can not explain this outcome, as it did not fit with the rest of our study's results. Therefore, this should be investigated more closely in future research.

7.2 Perceived workload and personal preference

While the previous section considered performance and established, that the controllers generally matched the baseline and in some cases improved on it, this section addresses the perceived workload connected to this performance, as well as the personal preferences of the participants.

The Raw TLX only showed a general tendency of ET causing a smaller workload. Looking at the separate categories, however, allows for a more specific analysis.
The categories mental demand, temporal demand and performance did not show significant differences, which basically fits the points made in the previous section. The controllers performed better in certain cases, but had similar performance as the baseline otherwise. For the entirety of the study, this meant, that the participants did not feel like the complexity of their tasks, the time they needed to complete the tasks, or their performance in the tasks significantly differed for any of the input methods.
In the remaining categories, physical demand, effort and frustration level, ET showed the lowest values with significant differences to both BL and HT. The lower physical demand and effort confirm our expectation of the reduced Gorilla Arm Syndrome for ET, which is probably due to the eye tracking enabling the user to always hold his arms in a comfortable position. In contrast to that, BL and HT required the user to constantly hold his hand, with or without a controller respectively, in front of his body.
The higher frustration in BL could have been caused by the problems some participants reported in the qualitative feedback, where they had to perform gestures multiple times, because they were not recognized or because rotary knobs could not be rotated far enough with one gesture. In HT, the higher frustration may have been due to having to operate the controller at the fixed positions of the UI elements, possibly leading to unpleasant interactions.

These differences in the perceived workload, especially the lower effort and frustration for ET, were also in line with most of the participants choosing ET as the best input method. The tight decision between BL and HT for second and third place fit to the TLX not indicating significant differences between BL and HT.

8 Summary and future work

In the course of this thesis, we built two cube-shaped controllers, with the goal of providing passive haptic feedback for the interaction with AR user interfaces. Subsequently, we evaluated them in a user study with 14 participants, which compared three input methods with each other. These methods were typical hand gestures, using a controller combined with hand tracking, and using a controller combined with eye tracking.

The results of this study mainly suggested two findings. First, adding passive haptic feedback in the form of a cube-shaped controller can improve the interaction with AR user interfaces performance-wise. This was the case for user interface elements like sliders and rotary knobs, especially ones with a high fidelity.
Second, both user preference and the perceived workload show an advantage for the controller combined with eye tracking, in comparison to the other two input methods.

All in all, out of the three input methods, the controller with eye tracking was the best one in our study.

8.1 Limitations

While the study fundamentally confirmed the validity of our approach to AR user interface interaction, there were some aspects, that it could not cover.

Because of time constraints, the study's participants only had around thirty minutes of actual AR user interface interaction. This time included all three input methods. Therefore, we could not tell, how the input methods compare in prolonged usage.

As for the participants themselves, all of them had either no, or only slight experience with augmented reality. This means, that the results might differ, when testing a group of participants with more diverse prior experiences.

Finally, there were two limitations, which we already mentioned in the discussion in section 7.1. We did not have enough data to verify, whether the radio button oversteps caused by the controllers were specifically due to the touchpad and trackball, as we expect, or if they were due to the general approach of using cube-shaped controllers.
Also, we could not explain the outcome of the controller with hand tracking being the faster method for high fidelity rotary knobs, but the slower one for high fidelity sliders.

8.2 Future Work

While our study evaluated the input methods in short-term usage, a large-scale user study would be beneficial. This would enable you to tell, whether prolonged usage reinforces the differences we found, especially when it comes to the Gorilla Arm Syndrome and the physical demand in general.

A larger scale could not only be applied to the temporal dimension of the study, but also to the hardware. The controllers created in this thesis contained five different input modalities. Further research could be done investigating more types of input modalities, ideally covering a large portion of Buxton's taxonomy[7]. By doing so, you could also directly compare the different input modalities against each other, in order to identify the best ones for different use cases.
With the information about which input modalities work best for which use cases, you could then also research the value of a modular controller system. Such a controller could feature freely interchangeable faces, allowing to adjust the available input modalities to your specific use case.
Another interesting aspect regarding the hardware could be active haptic feedback. By comparing controllers, similar to ours, with controllers with added active haptic feedback capabilities, for example through a vibration motor, it could be examined, whether active haptic feedback is a further improvement when it comes to the interaction with AR user interfaces.

At last, in contrast to our general study, you could also conduct further research with a specific area of application in mind. If you are focussing on industrial applications, for example, you could test participants after some practice with the input methods, as factory workers could be using them in their everyday work.

Bibliography

[1] Lonni Besançon et al. "Mouse, Tactile, and Tangible Input for 3D Manipulation". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 4727–4740. ISBN: 9781450346559. DOI: 10.1145/3025453.3025863.

[2] Mark Billinghurst, Tham Piumsomboon, and Huidong Bai. "Hands in Space: Gesture Interaction with Augmented-Reality Interfaces". In: *IEEE Computer Graphics and Applications* 34.1 (2014), pp. 77–80. DOI: 10.1109/MCG.2014.8.

[3] Florian Block et al. "Towards a Playful User Interface for Home Entertainment Systems". In: vol. 3295. Nov. 2004, pp. 207–217. ISBN: 978-3-540-23721-1. DOI: 10.1007/978-3-540-30473-9_20.

[4] Doug Bowman et al. "Using pinch gloves for both natural and abstract interaction techniques in virtual environments". In: *Proceedings of HCI International* (Jan. 2001).

[5] Evren Bozgeyikli and Lal Lila Bozgeyikli. "Evaluating Object Manipulation Interaction Techniques in Mixed Reality: Tangible User Interfaces and Gesture". In: *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 2021, pp. 778–787. DOI: 10.1109/VR50410.2021.00105.

[6] James V. Bradley. "Complete Counterbalancing of Immediate Sequential Effects in a Latin Square Design". In: *Journal of the American Statistical Association* 53.282 (1958), pp. 525–528. ISSN: 01621459. URL: http://www.jstor.org/stable/2281872.

[7] William Buxton. "Lexical and Pragmatic Considerations of Input Structures". In: *SIGGRAPH Comput. Graph.* 17.1 (Jan. 1983), pp. 31–37. ISSN: 0097-8930. DOI: 10.1145/988584.988586.

[8] Marcio C. Cabral, Carlos H. Morimoto, and Marcelo K. Zuffo. “On the Usability of Gesture Interfaces in Virtual Reality Environments”. In: *Proceedings of the 2005 Latin American Conference on Human-Computer Interaction*. CLIHC ’05. Cuernavaca, Mexico: Association for Computing Machinery, 2005, pp. 100–108. ISBN: 1595932240. DOI: 10.1145/1111360.1111370.

[9] Ken Camarata et al. “Navigational Blocks: Navigating Information Space with Tangible Media”. In: *Proceedings of the 7th International Conference on Intelligent User Interfaces*. IUI ’02. San Francisco, California, USA: Association for Computing Machinery, 2002, pp. 31–38. ISBN: 1581134592. DOI: 10.1145/502716.502725.

[10] Yun Suk Chang et al. “Evaluating gesture-based augmented reality annotation”. In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. 2017, pp. 182–185. DOI: 10.1109/3DUI.2017.7893337.

[11] Shuwei Chen et al. “Lower Limb Balance Rehabilitation of Post-stroke Patients Using an Evaluating and Training Combined Augmented Reality System”. In: *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 2020, pp. 217–218. DOI: 10.1109/ISMAR-Adjunct51615.2020.00064.

[12] Che Samihah Che Dalim et al. “TeachAR: An Interactive Augmented Reality Tool for Teaching Basic English to Non-native Children”. In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 2016, pp. 344–345. DOI: 10.1109/ISMAR-Adjunct.2016.0113.

[13] J Divya Udayan et al. “Augmented Reality in Brand Building and Marketing – Valves Industry”. In: *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. 2020, pp. 1–6. DOI: 10.1109/ic-ETITE47903.2020.425.

[14] Barrett Ens et al. “Multi-Scale Gestural Interaction for Augmented Reality”. In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. SA ’17. Bangkok, Thailand: Association for Computing Machinery, 2017. ISBN: 9781450354103. DOI: 10.1145/3132787.3132808.

[15] Wooi Boon Goh et al. “The I-Cube: Design Considerations for Block-Based Digital Manipulatives and Their Applications”. In: *Proceedings of the Designing Interactive Systems Conference*. DIS ’12. Newcastle Upon Tyne, United Kingdom: Association for Computing Machinery, 2012, pp. 398–407. ISBN: 9781450312103. DOI: 10.1145/2317956.2318016.

[16] Azhar Habib and Ahmed Hakim. "Context aware augmentational marketing". In: *2016 SAI Computing Conference (SAI)*. 2016, pp. 1227–1231. doi: 10.1109/SAI.2016.7556135.

[17] Jeff Hansberger et al. "Dispelling the Gorilla Arm Syndrome: The Viability of Prolonged Gesture Interactions". In: July 2017, pp. 505–520. isbn: 978-3-319-57986-3. doi: 10.1007/978-3-319-57987-0_41.

[18] Sandra G. Hart. "Nasa-Task Load Index (NASA-TLX); 20 Years Later". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50.9 (2006), pp. 904–908. doi: 10.1177/154193120605000909.

[19] Sandra G. Hart and Lowell E. Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research". In: *Human Mental Workload*. Ed. by Peter A. Hancock and Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. doi: https://doi.org/10.1016/S0166-4115(08)62386-9. url: https://www.sciencedirect.com/science/article/pii/S0166411508623869.

[20] Juan David Hincapié-Ramos et al. "Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 1063–1072. isbn: 9781450324731. doi: 10.1145/2556288.2557130.

[21] Hiroshi Ishii and Brygg Ullmer. "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms". In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: Association for Computing Machinery, 1997, pp. 234–241. isbn: 0897918029. doi: 10.1145/258549.258715.

[22] Florian Kern et al. "Using Hand Tracking and Voice Commands to Physically Align Virtual Surfaces in AR for Handwriting and Sketching with HoloLens 2". In: *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. VRST '21. Osaka, Japan: Association for Computing Machinery, 2021. isbn: 9781450390927. doi: 10.1145/3489849.3489940.

[23] Daniel Kharlamov et al. "TickTockRay: Smartwatch-Based 3D Pointing for Smartphone-Based Virtual Reality". In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. VRST '16. Munich, Germany: Association for Computing Machinery, 2016, pp. 365–366. isbn: 9781450344913. doi: 10.1145/2993369.2996311.

[24] Yang Kuang and XiaoMei Bai. “The Feasibility Study of Augmented Reality Technology in Early Childhood Education”. In: *2019 14th International Conference on Computer Science Education (ICCSE)*. 2019, pp. 172–175. DOI: 10.1109/ICCSE.2019.8845339.

[25] Kevin Lefeuvre et al. “Bricks, Blocks, Boxes, Cubes, and Dice: On the Role of Cubic Shapes for the Design of Tangible Interactive Devices”. In: *Proceedings of the 2018 Designing Interactive Systems Conference*. DIS ’18. Hong Kong, China: Association for Computing Machinery, 2018, pp. 485–496. ISBN: 9781450351980. DOI: 10.1145/3196709.3196768.

[26] R.W. Lindeman, J.L. Sibert, and J.K. Hahn. “Hand-held windows: towards effective 2D interaction in immersive virtual environments”. In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. 1999, pp. 205–212. DOI: 10.1109/VR.1999.756952.

[27] Diako Mardanbegi and Thies Pfeiffer. “EyeMRTK: A Toolkit for Developing Eye Gaze Interactive Applications in Virtual and Augmented Reality”. In: *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ETRA ’19. Denver, Colorado: Association for Computing Machinery, 2019. ISBN: 9781450367097. DOI: 10.1145/3317956.3318155.

[28] Andrii Matviienko et al. “CubeLendar: Design of a Tangible Interactive Event Awareness Cube”. In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 2601–2608. ISBN: 9781450340823. DOI: 10.1145/2851581.2892278.

[29] Dominic Potts, Martynas Dabravalskis, and Steven Houben. “TangibleTouch: A Toolkit for Designing Surface-Based Gestures for Tangible Interfaces”. In: *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’22. Daejeon, Republic of Korea: Association for Computing Machinery, 2022. ISBN: 9781450391474. DOI: 10.1145/3490149.3502263.

[30] Radiah Rivu et al. “StARe: Gaze-Assisted Face-to-Face Communication in Augmented Reality”. In: *ACM Symposium on Eye Tracking Research and Applications*. ETRA ’20 Adjunct. Stuttgart, Germany: Association for Computing Machinery, 2020. ISBN: 9781450371353. DOI: 10.1145/3379157.3388930.

[31] Young K. Ro, Alexander Brem, and Philipp A. Rauschnabel. “Augmented Reality Smart Glasses: Definition, Concepts and Impact on Firm Value Creation”. In: *Augmented Reality and Virtual Reality: Empowering Human, Place and Business*. Ed. by Timothy Jung and M. Claudia tom Dieck. Cham: Springer International Publishing,

2018, pp. 169–181. ISBN: 978-3-319-64027-3. DOI: 10.1007/978-3-319-64027-3_12.

[32] Anne Roudaut et al. “Rubikon: A Highly Reconfigurable Device for Advanced Interaction”. In: *CHI ’14 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 1327–1332. ISBN: 9781450324748. DOI: 10.1145/2559206.2581275.

[33] Orit Shaer and Eva Hornecker. “Tangible User Interfaces: Past, Present, and Future Directions”. In: *Foundations and Trends in Human-Computer Interaction* 3 (Jan. 2009), pp. 1–137. DOI: 10.1561/1100000026.

[34] J.G. Sheridan et al. “Exploring cube affordance: towards a classification of non-verbal dynamics of physical interfaces for wearable computing”. In: *2003 IEE Eurowearable*. 2003, pp. 113–118. DOI: 10.1049/ic:20030156.

[35] Nikolaos Sidorakis, George Alex Koulieris, and Katerina Mania. “Binocular eye-tracking for the control of a 3D immersive multimedia user interface”. In: *2015 IEEE 1st Workshop on Everyday Virtual Reality (WEVR)*. 2015, pp. 15–18. DOI: 10.1109/WEVR.2015.7151689.

[36] Lucia Terrenghi et al. “A Cube to Learn: A Tangible User Interface for the Design of a Learning Appliance”. In: *Personal Ubiquitous Comput.* 10.2–3 (Jan. 2006), pp. 153–158. ISSN: 1617-4909. DOI: 10.1007/s00779-005-0025-8.

[37] Takumi Toyama et al. “A Mixed Reality Head-Mounted Text Translation System Using Eye Gaze Input”. In: *Proceedings of the 19th International Conference on Intelligent User Interfaces*. IUI ’14. Haifa, Israel: Association for Computing Machinery, 2014, pp. 329–334. ISBN: 9781450321846. DOI: 10.1145/2557500.2557528.

[38] Rong Wen et al. “Hand gesture guided robot-assisted surgery based on a direct augmented reality interface”. In: *Computer Methods and Programs in Biomedicine* 116.2 (2014). New methods of human-robot interaction in medical practice, pp. 68–80. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2013.12.018. URL: https://www.sciencedirect.com/science/article/pii/S0169260713004082.

[39] Dennis Wolf et al. “Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–10. ISBN: 9781450367080. DOI: 10.1145/3313831.3376876.

[40] Robert Xiao et al. “MRTouch: Adding Touch Input to Head-Mounted Mixed Reality”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (2018), pp. 1653–1660. DOI: 10.1109/TVCG.2018.2794222.