Interaction Techniques for Mid-Air Pen Input in Handheld Augmented Reality

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Philipp Wacker, M.Sc.

aus Steinfurt, Deutschland

Berichter: Professor Dr. Jan Borchers
Professor Dr. Rainer Malaka

Tag der mündlichen Prüfung: 26. Mai 2021

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek verfügbar.

Contents

List of Figures

List of Tables

Abstract

Augmented Reality changes the way we interact with virtual information. Currently, virtual information is shown on 2D screens, separated from the real world. With Augmented Reality, virtual content can be shown directly embedded in the real world. This opens up the area of *situated modeling* in which virtual models are designed in context of the real world to, for example, print them out using a 3D printer. In an initial study, we show that sketching on physical objects improves stroke accuracy compared to strokes on virtual objects, and that features guiding a stroke, either through a concave or convex shape or through a visual guide, further improve the accuracy especially for physical objects.

The most available form of Augmented Reality (AR) is Handheld Augmented Reality which shows the virtual information embedded in the camera view of everyday smartphones or tablets. However, continuously specifying a 3D position—needed, e.g., for drawing in mid-air—is not directly possible in today's systems. We build the ARPen system to allow for *situated modeling* in Handheld AR, requiring only a 3D-printed pen and a consumer smartphone. But many essential interactions are not yet clear for such a bimanual system. We design and evaluate *selection & manipulation techniques* to adjust the pose of a mid-air object, as well as *menu techniques* to control properties of objects in the scene. We show that ray-casting techniques, especially through the tip of the pen, generally perform well. However, interacting on the touchscreen or even combinations of both touchscreen and mid-air input also achieve promising results. To overcome perception issues of determining the depth of virtual objects in Handheld AR, we design *depth visualizations* that show the position of the pen tip in relation to other objects in the scene. We identify that a *heatmap* visualization, coloring every object in the scene depending on their distance to the pen tip, achieves best results and was preferred by study participants.

We release the ARPen system as an open-source toolbox, enabling researchers to implement and evaluate interaction techniques for Handheld AR with a mid-air pen. Our findings on essential interaction techniques provide a starting point for the development and evaluation of specialized application scenarios.

Überblick

Augmented Reality (AR) verändert die Art der Interaktion mit virtuellen Informationen. Derzeit werden virtuelle Informationen getrennt von der realen Welt auf 2D-Bildschirmen angezeigt. Mit AR können virtuelle Inhalte direkt in die reale Welt eingebettet werden. Dies ermöglicht das *Vor-Ort-Modellieren*, wobei virtuelle Modelle im Kontext der realen Welt entworfen werden. In einer ersten Studie wird gezeigt, dass das Zeichnen auf physischen Objekten die Strichgenauigkeit im Vergleich zu Strichen auf virtuellen Objekten verbessert und dass Merkmale, die einen Strich durch eine konkave/konvexe Form oder durch eine visuelle Linie führen, die Genauigkeit weiter erhöhen.

Die verbreitetste Form von AR ist Handheld AR, bei der virtuelle Informationen in das Kamerabild von alltäglichen Smartphones oder Tablets eingebettet werden. Jedoch ist die kontinuierliche Angabe einer 3D-Position, die z.B. für das Zeichnen in der Luft benötigt wird, in heutigen Systemen nicht direkt möglich. Das ARPen-System wird vorgestellt, welches *Vor-Ort-Modellieren* in Handheld AR mithilfe eines 3D-gedruckten Stiftes und eines gängigen Smartphones ermöglicht. Die wesentlichen Interaktionen für ein solches bimanuelles System sind noch nicht ausreichend entwickelt. Es werden *Auswahl- und Manipulationstechniken* zur Positionierung eines Objekts in der Luft entworfen und evaluiert. Darüber hinaus werden *Menütechniken* zur Steuerung von Objekteigenschaften in der Szene entwickelt. Es wird gezeigt, dass RayCasting-Techniken, insbesondere durch die Stiftspitze, gut funktionieren. Auch die Interaktion über den Touchscreen oder Kombinationen aus Touchscreen- und Stift-Eingabe erzielen vielversprechende Ergebnisse. Um Wahrnehmungsprobleme bei der Bestimmung der Tiefe von virtuellen Objekten in Handheld AR zu überwinden, werden *Tiefenvisualisierungen* entworfen, die die Position der Stiftspitze in Relation zu anderen Objekten in der Szene zeigen. Eine *Heatmap*-Visualisierung, die jedes Objekt in der Szene in Abhängigkeit von seinem Abstand zur Stiftspitze einfärbt, erzielt die besten Ergebnisse und wird von den Studienteilnehmern bevorzugt.

Das ARPen-System wird als eine Open-Source-Toolbox veröffentlicht, die es ermöglicht, Interaktionstechniken für Handheld AR mit einem Mid-Air-Stift zu implementieren und zu evaluieren. Die erzielten Erkenntnisse über wesentliche Interaktionstechniken bieten einen Ausgangspunkt für die Entwicklung und Evaluierung von spezialisierten Anwendungsszenarien.

Acknowledgements

This PhD thesis would not have been possible without the support of many people.

First, I want to thank Prof. Dr. Jan Borchers for giving me the opportunity to pursue my PhD in Human-Computer Interaction and for providing a work environment in which it was possible to find and follow my research interests. He provided invaluable support from guiding research ideas to fine-tuning publications. I also want to thank the second supervisor of this thesis, Prof. Dr. Rainer Malaka, for spending the time and effort to co-supervise this thesis and giving me the opportunity to present my research at the graduate school of his Digital Media Lab.

I started work at a time when several colleagues were in the final stages of their PhD. Thank you to Florian Heller, Thorsten Karrer, Moritz Wittenhagen, Leonhard Lichtschlag, Jan-Peter Krämer, and especially Chat Wacharamanotham for passing on your knowledge during our time together. My research would not have been possible without my colleagues, in particular, Simon Völker and Christian Cherek. Thank you for the intense discussions and valuable feedback. My gratitude also goes to the students who have either joined me in researching AR interactions for their theses, or worked together with me on other projects and lectures. A special thanks to all participants who took part in the user studies required for this thesis. The time commitment and feedback is greatly appreciated.

The life outside of the work environment is equally important for the success of this PhD. I want to thank, in particular, Stefan and Christian for our gaming nights and trips through Germany and Jan and Kathrin for welcome distractions from work whenever we were in the same area. Thank you to Jan, Torsten, and Stefan for proofreading parts of this thesis and providing indispensable feedback.

Most of all, I want to thank Michaela, Mechthild, Peter, Bastian, and Tanja as well as Gabi and Robert. Without your constant support, help, warmth, enthusiasm, and everything else, none of this would have been possible.

Conventions

Throughout this thesis, we use the following conventions:

- The thesis is written in American English.
- The first person is written in plural form.
- Unidentified third persons are described in female form.

Margin notes at the side of the page are used to summarize important parts of a paragraph, where applicable. Summary of paragraph.

Names of software frameworks are written in `typewriter-style text`.

In cases where content was published both as a student thesis under the guidance of the author of this thesis and as a peer-reviewed paper at a conference, the paper is given preference in citations. The thesis is then referenced at the beginning of the appropriate chapter.

Chapter 1

Introduction

Today's interaction with virtual media mostly takes place in two dimensions. Devices ranging from large scale televisions to small smartwatches visualize their information on 2D displays and the interaction with content is also performed by specifying 2D information—either through touching the screen or through pointing devices that descend from the first mouse presented by Engelbart and English [1968]. Every visualization of a third dimension, for example by windows overlapping each other in current desktop operating systems, is only an illusion since the content is still fixed to the 2D screen. Nielsen [1993] calls these *2.5-D Interfaces*. Virtual 3D environments, such as Augmented Reality (AR) and Virtual Reality (VR), make it possible to lift the information in the 3rd dimension and visualize it in mid-air. This means that the digital information is no longer anchored to a 2D position on the display but to a 3D position in the environment around the user (cf. Figure 1.1, left). An example is a virtual representation of the rainforest that the observer can walk through to learn about the ecosystem (cf. Figure 1.1, right).

Currently, virtual information is displayed mostly on 2D screens.

Augmented and Virtual Reality visualize content in 3D around the user.

But seeing objects in mid-air is only one aspect of the possibilities of such systems. We also want to further interact in this environment and with the objects in it. Evaluating this interaction in Augmented Reality, and Handheld AR in particular, is the main topic of this thesis.

In this thesis, we investigate interactions in Augmented Reality.

Figure 1.1: *Left*: A head-mounted Augmented Reality display showing virtual objects embedded in the real world (Image: [Microsoft, 2016]). *Right*: A virtual rainforest is included in the camera view of a tablet using the WWF Forests App[1].

Currently, 3D objects for VR and AR are created mostly on 2D systems. Designers prefer sketching their ideas in 2D on paper.

A specific type of interaction is the creation of additional virtual objects. Currently, most of the 3D models in virtual environments are created on 2D computer displays with software such as Blender[2] [Schmalstieg and Höllerer, 2016, p. 312]. Many projects have investigated how 3D models can be created from 2D drawings, since particularly in the early design stages, many professionals prefer sketching their ideas on paper. Olsen et al. [2009] provide a survey of techniques and challenges for projects that use sketches for 3D modeling of objects. An important point they make is that sketching has a divergent user base ranging from artists to engineers, who all have their own motivations and styles of sketching, resulting in different technical needs.

Many projects investigate how 2D sketches can be used for 3D modeling.

The origins of sketching on computer-based systems can be traced back to Sutherland [1963]. With his *Sketchpad* system, users could draw lines on a computer screen and even create simple 3D models. Today, many approaches interpret 2D sketches on a tablet to create new 3D geometries. To address the issue of the missing 3rd dimension when creating 3D objects on a 2D surface, these projects use different techniques. These techniques include sketching from different viewpoints [Igarashi et al., 1999; Bae et al., 2008], positioning strokes on different drawing planes [Grimm and Joshi, 2012], or adding graphical cues in the 2D sketch that the

[1] apps.apple.com/us/app/wwf-forests/id1518039408 (accessed 28.05.2021)
[2] www.blender.org/ (accessed 28.05.2021)

system can interpret [Xu et al., 2014]. Another technique is to first create a scaffold of perspective lines so that the system can interpret following curved strokes relative to that scaffold [Schmidt et al., 2009]. Often, the designed object should have a connection with an already existing one. For this, projects such as those by Paoli and Singh [2015] and Schmid et al. [2011] interpret the strokes to 'wrap around' an existing 3D model, e.g., of a mannequin. To use the context provided by objects in a 2D image, Lau et al. [2010] interpret sketches on a photo relative to other objects in the image to infer a 3D model fitting into the scene.

Augmented and Virtual Reality allow for modeling objects directly in 3D.

If a 3D input device, for example in the shape of a pen, is tracked in mid-air, Augmented and Virtual Reality allow performing the modeling task directly in 3D. Such a "painting in mid-air" is a fascination that has been apparent in the domain of art in the past already. Picasso used a light pen to draw shapes and elements in mid-air (see [Cosgrove, 2012] for photos). These shapes are only visible as lines on long time exposure photographs, so while they have been drawn in mid-air, they are only visible from one perspective, however, and it is not possible to edit them afterward. With drawing in Augmented or Virtual Reality, artists could draw lines in mid-air, edit and view them from different positions. This form of *modeling in mid-air* has already been around since the early 90s [Sachs et al., 1991; Butterworth et al., 1992; Deering, 1995] and it requires observing the position and orientation of input devices in space. The movement of the input device can then be aligned with the visualization to create the impression of creating strokes in mid-air, as in the FreeDrawer system by Wesche and Seidel [2001], for example. To align with the process of preparing initial sketches in 2D on paper, Jackson and Keefe [2016] let users arrange digital scans of their 2D sketches in a virtual 3D environment. Then, users create a wireframe model by selecting and "lifting" individual strokes from those sketches, and pull out surfaces between the strokes. Today, products like the recently open-sourced Tilt Brush[3] application or Gravity Sketch VR[4] let users create and view models directly in 3D using commercially available Virtual Reality devices.

[3] www.tiltbrush.com (accessed 28.05.2021)
[4] www.gravitysketch.com/ (accessed 28.05.2021)

Figure 1.2: *Left*: Screenshot of the VR game *Beat Saber* in which the player has to cut boxes to the rhythm of the music (Photo by Matthias Heetkamp). *Right*: Illustration of the game *Minecraft* viewed through an AR headset (Image: [Microsoft, 2015]).

In VR, the environment is virtual; in AR, virtual elements are added into the real world.

The type of virtual environment is an important factor for these modeling tasks. In Virtual Reality, the surrounding environment is digitally generated while in Augmented Reality, the majority of the environment is the real world and only some virtual elements are rendered to appear as if in the real world [Milgram and Kishino, 1994; Azuma, 1997] (cf. Figure 1.2). This means that in Augmented Reality, real, physical objects can also be used during the modeling process. This presents a big possibility for areas such as Personal Fabrication in which users use digital fabrication tools such as lasercutters and 3D printers to realize their own projects [Gershenfeld, 2007; Mota, 2011]. While 3D printers can now be purchased at hardware stores and 3D models of many objects are available online, it requires further knowledge to modify these models for a specific use case or to create a new model for which you have only the idea in your head. Examples of such ideas could be a lampshade for a lamp socket, a fitting attachment for a flower pot, or an inset for the cup holder in a car to secure smaller sized cans. With drawing and designing in Augmented Reality, it is possible to design the 3D models directly in the location in which they are meant to be used before printing them out on a 3D printer. Schmalstieg and Höllerer [2016, p. 312] use the term *situated modeling* for this interaction and it would enable even non-professional designers to quickly create simple 3D models fitting existing objects in the world. For the example of the cup holder inset, this means that a user could take the AR drawing tool into

The process of creating models directly in the real world is called *situated modeling*.

Figure 1.3: Smaller sized cans are often loose in cup holders *(a)*. With *situated modeling*, the user can trace the environment to create simple shapes *(b)* and combine them *(c)* to create an inset that fits both the cup holder and the can *(d)*.

the car, trace the shape of the existing cupholder and the can she wants to fit, before combining them to create the model of an inset that matches both the cup holder and the can (cf. Figure 1.3). While the drawing operation would also be possible in Virtual Reality, it would be necessary to digitize the relevant parts of the physical environment beforehand—and even then, the haptic feedback of tracing over a real object would be missing.

But how much does the ability to draw on physical objects affect the drawing performance? To figure this out, the first part of this thesis compares drawing performance under different conditions. We found that drawing on physical objects increases the accuracy of lines drawn by $\approx$ 50 %. Further differences in the surface of the object, such as a concave or convex shape, influence the precision to a more minor degree.

Drawing on a physical object increases the accuracy by $\approx$ 50 % compared to drawing on virtual objects.

Another important question is, how the benefits of modeling and interacting in AR can be made more available to everyone without the need for expensive and professional equipment. To view the virtual objects in the real world, users need either a special set of glasses (head-mounted display, HMD; cf. Figure 1.2, right) or a handheld device, such as a smartphone, that acts as a window to see the virtual objects (Handheld AR) (cf. Figure 1.4). While a few head-mounted devices have been released over the past years, such as Microsoft's HoloLens[5] or the Magic Leap 1[6], Handheld AR is the most used form of AR to date because

To see objects in AR, users need either a head-mounted display (HMD) or a handheld display (Handheld AR).

[5] www.microsoft.com/en-us/hololens (accessed 28.05.2021)
[6] www.magicleap.com/en-us (accessed 28.05.2021)

Figure 1.4: In Handheld AR, virtual information is incorporated into the live camera view of mobile devices such as smartphones and tablets (Image: [Apple, 2020]).

We focus our evaluations on Handheld AR.

of its support in major mobile operating systems through frameworks such as `ARKit`[7] and `ARCore`[8] [Schmalstieg and Höllerer, 2016, p. 69]. Because of this widespread availability of Handheld AR and its interesting research opportunities, we focus on interactions in Handheld AR for the majority of this thesis.

Continuous 3D input, e.g., to draw lines, is not directly supported in Handheld AR.

However, continuous 3D input required for drawing lines in mid-air with six degrees of freedom, is not directly supported in Handheld AR. To overcome this limitation and enable *situated modeling* in Handheld AR, Napkin Sketch by Xin et al. [2008], lets a user specify a drawing plane on a tablet in relation to a "napkin" displayed in AR in front of the user. The system then maps the 2D strokes on the tablet directly onto that plane and visualizes the resulting model in AR, on top of the live camera image. Huo et al. [2017] map touch strokes onto surfaces tracked in the real world, and allows touchscreen-based manipulation of the resulting objects. To enable 3D input for Handheld AR with six degrees of freedom, we developed the *ARPen system* allowing mid-air pen interaction with a custom 3D-printed pen or even just a pen printed with a normal printer. The system presents a toolbox for developers to try out mid-air pen interaction in Handheld AR and is an easy way to implement and test new interaction techniques. We found

We developed the ARPen system to enable 3D input using a consumer smartphone and a 3D-printed pen.

[7] developer.apple.com/augmented-reality/ (accessed 28.05.2021)
[8] developers.google.com/ar (accessed 28.05.2021)

that many questions how to perform basic interactions using a Handheld AR system with a mid-air pen have not been answered yet, so we evaluate these essential interactions in this thesis, instead of focussing on the *situated modeling* application scenario alone. However, we keep *situated modeling* in mind throughout the evaluations both to focus design decisions and to have a consistent example scenario.

We design and evaluate basic interaction techniques for Handheld AR with a mid-air pen.

The interactions we look at follow the classification from LaViola et al. [2017]. They state that interactions in 3D user interfaces can be categorized into three groups. The first group contains *selection & manipulation* tasks, such as moving an object to a different location or changing its size. The second group is focussed on controlling the overall system, for example, by using context menus, and they call this group *system control*. The third group is called *navigation*. This includes the motor component of *travel*—*how* to move through the virtual environment—but also the cognitive component of *wayfinding*—*where* to move to in the virtual environment. The last component is particularly relevant for Handheld AR as this relates to understanding the spatial relationship between the input device and virtual elements in the scene—a frequent problem due to perception issues in Handheld AR [Kruijff et al., 2010].

We structure our interaction techniques into *selection & manipulation*, *system control*, and *navigation* tasks.

In summary, the contributions made through this thesis are:

1. A study on the influences of physical objects and different surface conditions on sketching accuracy in Augmented Reality.
2. The ARPen system as a toolbox for evaluations in Handheld AR with a mid-air pointing device. The code of the system is available open-source on Github[9]. The interpretation of the mid-air position is managed through a plugin system so that researchers can easily add their own interactions. We also incorporated the modeling kernel `OpenCascade` to enable the creation and interaction with complex geometries. A stable version highlighting the different interaction techniques is also available in the iOS App Store[10].

[9] github.com/i10/ARPen (accessed 28.05.2021)
[10] hci.rwth-aachen.de/arpen-ios (accessed 28.05.2021)

3. Evaluations of interaction techniques for common 3D *selection & manipulation* tasks in Handheld AR with a mid-air pen. Using the ARPen system, we looked at:
   - Selection Techniques, to pick a specific object from the scene.
   - Translation Techniques, to move an object to a new location.
   - Rotation Techniques, to change the orientation of an object.
   - Scaling Techniques, to increase or decrease the size of an object.
4. Evaluation of different menu styles as an example for *system control* tasks. Menus are an essential interaction part as they allow the grouping of different actions that can be used to adjust the whole system or the properties of a specific object.
5. Evaluation of visualization techniques to clarify the position of the pen in relation to the virtual objects in the scene. This visualization is needed as it is difficult to perceive the depth of projected objects onto the screen of the smartphone.

1.1 Structure

Following this introduction, this thesis consists of the following chapters:

Chapter 2 Here, we provide a brief introduction into the history of Augmented Reality, the major technological components of AR, and an overview of perceptual issues in Augmented Reality.

Chapter 3 We first look at the potential of interacting in Augmented Reality by investigating the impact of drawing on physical objects compared to drawing on virtual objects.

Chapter 4 Encouraged by the results, we designed and implemented the ARPen system: A Handheld AR system that tracks a pen in mid-air for 3D input. This presents a cost-efficient way of providing 3D input in Handheld AR. Having a smartphone as the window into the AR world opens interesting questions regarding the interaction with such a system.

Chapter 5 To better understand the range of interactions required in 3D User Interfaces, we provide an overview of the categories by LaViola et al., 2017. These tasks range from *selection & manipulation* tasks, over *system control* tasks such as menus, to *wayfinding* visualizations to address depth perception issues in such a system.

Chapter 6 This chapter contains studies on basic object *selection & manipulation* tasks. This includes in-depth evaluations of selection (Chapter 6.2) and translation (Chapter 6.3) as well as overviews of techniques for rotation (Chapter 6.4.1) and scaling (Chapter 6.4.2).

Chapter 7 To investigate *system control* tasks, we evaluated different menu styles for Handheld AR with a mid-air pen. This chapter contains details on our study and a presentation of the results.

Chapter 8 Perceiving the correct depth location of objects in Handheld AR is difficult. In this chapter, we present our study on different visualization techniques to show the position of the mid-air pen in relation to other objects in the scene.

Chapter 9 To finish this thesis, we sum up all our contributions and present future directions to continue with Handheld AR with a mid-air pen.

Appendix A Throughout the thesis, we evaluated our results using confidence intervals and estimation. As these techniques are currently not common in the HCI community, we provide a brief description of these techniques.

In the next chapter, we introduce important definitions and concepts of Augmented Reality, such as different display and tracking types.

Chapter 2

Overview of Augmented Reality

"Augmented Reality isn't going to be a big thing—it's going to be everything"

—Jon Peddie in his book about Augmented Reality

Similar to Peddie [2017], Schmalstieg and Höllerer [2016] see Augmented Reality (AR) as the next step in the line of developments that increase the availability of information. These developments include the creation of the *world wide web* that enables everyone to access a multitude of information, the *social web* that allows people to connect and share content, to the rise of *smartphones* that provide access to this content almost everywhere. However, currently users have to take out their phone and access information on a screen detached from the real world. With Augmented Reality, this information becomes embedded in the real world and can be seen through glasses or handheld displays.

Augmented Reality increases the availability of information.

In this chapter, we provide an overview of Augmented Reality to introduce concepts that are used throughout this thesis. First, we are defining Augmented Reality and report important milestones of the field (Chapter 2.1) before briefly explaining the different technological components of an AR system (Chapter 2.2). The information in both these chapters is based on the book "Augmented Reality – Principles and Practice" by Schmalstieg and Höllerer [2016] and we recommend this book for further details and aspects not covered here. In the final part of this chapter, we address perceptual issues in Augmented Reality that affect the interaction with such systems (Chapter 2.3).

2.1 Definition and History

Augmented Reality (AR) describes the embedding of virtual artifacts in the real world. To distinguish it from the area of Virtual Reality (VR) and special effects in movies, Azuma [1997] notes three characteristics in his often used definition of AR:

Azuma [1997] specifies 3 characteristics of AR.

1. AR "combines real and virtual",
2. AR is "interactive in real time",
3. AR is "registered in 3D".

The first part of this definition separates AR from VR. In VR, the user is in a completely virtual environment while in AR both the real world and virtual elements are visible at the same time. The third part of the definition makes clear that these elements need to be embedded in the 3D world to distinguish AR from simple overlays such as camera controls on smartphone cameras. While visual effects in movies such as "The Lord of the Rings" merge virtual elements such as Trolls with real world recordings so that it seems as if both exist together, it does not fulfill the second part of the definition.

We focus on *visual* Augmented Reality.

Note that this definition of AR does not give any requirements to the technologies used to create AR and also not to the modalities that are augmented. In this thesis we focus on *visual* Augmented Reality, which means the inclusion of visual objects within the real world. However, *audio* Augmented Reality, placing virtual sounds in the real world, and to a lesser extent Augmented Reality involving smell

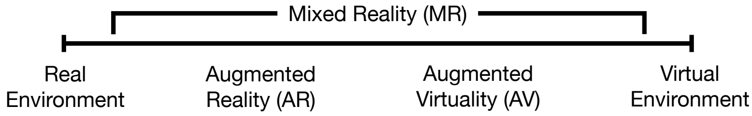


Figure 2.1: The Reality-Virtuality Continuum used to define the area of Mixed Reality between the end points of the real and virtual environment (adapted from [Milgram and Kishino, 1994]).

and taste, also exist and are subject of research. We refer to Schmalstieg and Höllerer [2016] for more information on these types of AR.

Another popular characterization of Augmented Reality is through the *Reality-Virtuality Continuum* by Milgram and Kishino [1994]. Figure 2.1 shows the continuum between the real world on side and a completely virtual world on the other. Introducing real objects in the virtual environment or virtual objects in the real environment places the resulting system along the continuum between the two ends, in the area that Milgram and Kishino name *Mixed Reality*. Using this characterization, *Augmented Reality* is located towards the real world which means that the main elements in the environment are still real but that virtual elements are embedded into it. Its mirror on the other side of the continuum is *Augmented Virtuality*, an environment in which most aspects are virtual but there exist links to the real world such as a window through which the real world can be seen.

The *Reality-Virtuality Continuum* describes the space between the real world and the virtual world.

Based on these definitions, we see Augmented Reality as the addition of virtual visual objects in an otherwise predominantly real environment in a way that they are both aligned with the real world as well as interactively controlled in real time.

Definition: *Augmented Reality*.

Following, we mention several milestones in the history of Augmented Reality. Details on different styles of Augmented Reality mentioned in this overview are described more clearly in Chapter 2.2.

Similar to the origin of sketching on a computer, the origin of both Augmented Reality and Virtual Reality can be

The origin of AR and VR can be traced back to Sutherland.

traced back to Sutherland [1965] who described an *ultimate display*. According to Sutherland, this ultimate display is able to represent every virtual object not only visually but also with all other modalities. This means that it would be possible to sit on a "virtual" chair created by the system. Three years later, Sutherland presented an immersive system nicknamed "The Sword of Damocles" that was already able to track the orientation of the head and show information with see-through displays. The whole system was suspended from the ceiling because of its weight.

Handheld systems were first developed in the mid 90s.

The term "Augmented Reality" was first used by Caudell and Mizell [1992] for an airplane manufacturing system in which the next steps for construction were shown with a head-mounted system. A first handheld system is the Chameleon-System by Fitzmaurice [1993]. This system consisted of a handheld display which position and orientation were tracked. The display showed the video stream of a camera which was pointed at a workstation screen. This screen displayed content based on the current position of the handheld device. This way, the Chameleon-System could, for example, show additional information about a city when it determined that the handheld display was positioned over a map at the location of a city. This system tracked the position of the screen in space but did not merge the information into a controllable camera view. Rekimoto and Nagao [1995] presented the first Handheld AR display that also used a camera on the back to enable seeing "through" the display. The position of virtual information was determined with the help of markers. However, this system still needed to be connected to a workstation for the calculations.

The release of ARToolkit made marker tracking simpler for researchers.

The release of the open-source ARToolkit by Kato and Billinghurst [1999] simplified the detection of black-and-white markers and the calculation of associated position information, making the use of marker tracking simpler for researchers. D. Wagner and Schmalstieg [2003] used marker tracking on a personal digital assistant to develop the first untethered Handheld AR system that can track markers in real time and show virtual information embedded in the camera view shown on the screen. In [2008], D. Wagner et al. presented with *natural feature tracking* a sys-

Figure 2.2: Modern cars show the driving path depending on the current position of the steering wheel in the video of a rearview camera (Photo by Mechthild Wacker).

tem that allows calculating the position of a mobile phone from a normal photo instead of a specific black-and-white marker. Removing prior knowledge of the environment entirely, methods such as *parallel tracking and mapping* (*PTAM*) [Klein and Murray, 2007] or *Kintect Fusion* [Newcombe et al., 2011] build a model of the environment while the system is moved around it.

In recent years, we have not only seen the release of standalone head-mounted AR devices such as Microsoft's HoloLens[1] or the Magic Leap 1[2] but also widespread support of AR frameworks in smartphone operating systems with `ARKit`[3] and `ARCore`[4], leading to an increasing number of applications using AR, such as the game Pokémon Go[5]. Other examples of Augmented Reality in use today are, for example, visualizations of the driving path in back facing cameras (cf. Figure 2.2). For more examples on AR systems in areas such as construction, maintenance, medicine, or navigation, we refer to Schmalstieg and Höllerer [2016, pp. 13–28].

Recent years have seen a wider adoption of AR through devices and frameworks.

[1] www.microsoft.com/en-us/hololens (accessed 28.05.2021)
[2] www.magicleap.com/en-us (accessed 28.05.2021)
[3] developer.apple.com/augmented-reality/ (accessed 28.05.2021)
[4] developers.google.com/ar (accessed 28.05.2021)
[5] pokemongolive.com/ (accessed 28.05.2021)

2.2 Technology

For an Augmented Reality system, there are two major questions that need to be answered:

1. *How* to show virtual information?
2. *Where* to show virtual information?

Displays are needed to show virtual objects in AR and *tracking* is needed to determine where to show them.

The first question relates to the selection of a *display* technology that needs to combine real and virtual worlds. The second question is about the choice of *tracking* technologies and algorithms. We present an overview of main properties for both questions, starting with *displays* in Chapter 2.2.1, followed by *tracking* in Chapter 2.2.2. Since the majority of this thesis is focussed on Handheld Augmented Reality, we point out how the technologies are applied for this type of AR system in particular. For detailed information, we refer to textbooks such as by Schmalstieg and Höllerer [2016].

2.2.1 Displays

The chosen display technology for an AR system determines how the user is able to see both the real world objects as well as virtual information embedded in the world. The display therefore has to be able to display virtual objects combined with the real world.

Displays can be *optical see-through* or *video see-through*.

The two major versions of displays are *optical see-through* and *video see-through* displays. In an *optical see-through* display, the virtual content is shown on a transparent display through which the user can still see the real environment (cf. Figure 2.3, left). On the other hand, for *video see-through* displays, a camera captures the real world and virtual information is introduced directly into the video feed (cf. Figure 2.3, right). A third version of AR display is *spatial projection* in which virtual content is projected directly on objects in the real world. Since this thesis focusses mostly on *optical see-through* and *video see-through* displays, we refer the

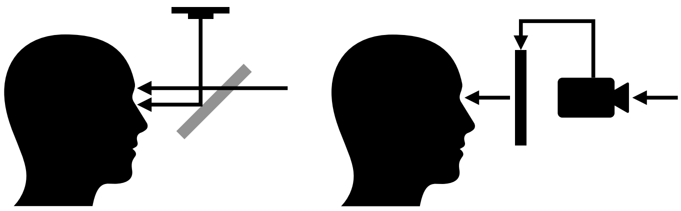


Figure 2.3: *Left*: In *optical see-through* displays, virtual information is displayed on a screen through which the real world is still visible. *Right*: In *video see-through*, the real world is captioned by a camera and virtual information is included in the video screen which is shown to the user. (Adapted from [Azuma, 1997]).

reader to Schmalstieg and Höllerer [2016, p. 42] for more information on *spatial projection*.

Displays can be *wearable* or *non-wearable*.

Another differentiating factor for AR displays is their placement in respect to the user. Peddie [2017] classifies AR systems into *wearable systems* such as *head-mounted displays* (*HMD*) that are worn similar to glasses (cf. Figure 1.2, right), and *non-wearable systems*, such as handheld displays or projector setups (cf. Figure 1.4). Augmented Reality on handheld displays, such as smartphones, is called *Handheld AR*, while *stationary* or *projected* AR displays show AR content on a larger, often static screen. An example for *stationary* AR is a virtual changing room, but similarly to *spatial projection*, we refer to Schmalstieg and Höllerer [2016, pp. 72–83] for further details on *stationary* and *projected* AR.

We use the term *Handheld AR* throughout this thesis.

Handheld AR is occasionally also called *mobile AR* or *smartphone AR*. However, developments in recent years have made untethered HMDs possible so that they are also mobile, and since tablets are also able to display AR content, the term *smartphone AR* is too limiting. Therefore, we use the term *Handheld AR* throughout this thesis. Following, we compare different combinations of display technologies and list advantages and disadvantages.

Head-mounted displays can be either optical see-through or video see-through. For a video see-through HMD, two cameras can be used to record individual videos for each eye (binocular). Alternative solutions can also use only one

Figure 2.4: Head-worn, video see-through Augmented Reality can, for example, be achieved by wearing a smartphone with a single camera before the eyes. Photo by Maurizio Pesce under cc-by-2.0.

camera and present the same image to both eyes (bi-ocular) or even just one image in front of the user (monocular) (cf. Figure 2.4). Current Handheld AR systems are mostly monocular video see-through devices, since they run on smartphones using only one camera to record the environment and present the augmented scene.

Handheld AR on smartphones is mostly monocular video see-through.

There is no universally "best" type of Augmented Reality system. For the choice between optical see-through and video see-through, both types have advantages and disadvantages. For optical see-through, the advantage is that the user is still able to see the real world normally through the display. However, displaying virtual information is generally affected by latencies which cause virtual objects to lag briefly behind. Video see-through, on the other hand, has more control over this since the virtual information is embedded directly into the video stream so that it is easier to align the virtual objects and control for latency. However, viewing the real world through a camera feed is problematic. Even if multiple cameras are used, it is currently not possible to measure the fine movements of our eyes in such a quality that the video can be adapted to recreate our experience of the real world. This means, that we are aware of looking through a camera.

There is no "best" type of AR system. All current systems have advantages and disadvantages.

For the choice between HMD or Handheld AR, the situation is similar. Head-mounted displays leave the hands of the user free for interaction and often produce stereoscopic visualizations, which means that each eye receives its own image to simulate the depth position of virtual objects. On the other hand, current HMDs are only able to show AR in a more limited field of view compared to the normal field of view of a human. For Handheld AR systems, the area in which to see virtual objects is also limited. Additionally, ergonomics are an issue since it becomes strenuous to hold up a device for a prolonged period of time. Nevertheless, due to the widespread availability of smartphones, "handheld displays [are] the most popular platform for AR to date" [Schmalstieg and Höllerer, 2016, p. 69].

Handheld AR is the most available form of AR today.

2.2.2 Tracking

The role of tracking in an Augmented Reality system is the monitor the components of the system and elements of the environment. An understanding of these properties is needed so that the AR system is able to display virtual content at the appropriate position, creating the appearance as if the object is really in the real world while the user is moving around it. The position of elements in the scene and their relationship is described through transformations.

Tracking is needed to display virtual content at the correct position.

Schmalstieg and Höllerer [2016] mention three specific transformations for AR systems. The first, the *model transform*, is used to describe the position and orientation of objects in the scene—often also referred to as the *pose* of the object. The *view transform* represents the position of the camera in the scene. Lastly, the *perspective transform* describes how the tracking information is interpreted to show virtual information on the display. This final transform is usually determined offline as it does not change during the use of the system. For example, the *perspective transform* in video see-through Handheld AR defines how the video stream from the camera is shown on the screen.

3 transforms are needed for AR systems.

In a completely static system, the other two transforms can also be calibrated beforehand. However, in order to have

Tracking is the process of updating transforms in real-time.

a dynamic system, at least one of the transforms needs to be adapted in real-time. *Tracking* is this process of determining the updated transforms. Being able to update the *view transform* enables the user to move through the environment and have virtual information remain at their real world location. Being able to track the *model transform* of a real world object means that virtual information attached to it would move with the real world object as it is moved around the environment.

The required level of tracking is dependent on the *frames of reference*.

How virtual objects are attached in the real world, is defined by their *frame of reference*. Schmalstieg and Höllerer [2016, pp. 89–90] mention different frames of reference, first defined by Feiner et al. [1993]. Adapted from Feiner et al., a virtual object can be *world-stabilized*, its position in the environment is fixed, *object-stabilized*, its position is linked to another object and if the object moves, the virtual object moves with it, or *screen-stabilized*, the virtual object is always shown at the same location on the screen. A special case of *object-stabilized* is *body-stabilized*. This describes objects that move with the user, such as an informational window always floating to the left of the user. The types of frames of reference required in an application determine which tracking of transforms is needed. Having only *screen-stabilized* objects requires no further tracking. *World-stabilized* objects need keep their position as the user moves around the world so that tracking the *view transform* is necessary. Finally, if virtual information should also "stick" to a physical object while it is moved around the scene, tracking the *model transform* for this object is also required.

Tracking systems can be *outside-in* or *inside-out* systems.

Tracking systems generally fall into one of two categories: *outside-in* or *inside-out* tracking. These categories differ in the placement of the sensors and *what* is tracked by the sensors. In an *outside-in* tracking system, the sensors are placed in the environment and track the elements in the scene (cf. Figure 2.5, left). While this requires preparation—and also limitation—of the interaction space, using multiple sensors placed in the environment allows tracking from different positions. Therefore, the measurements can be used to both triangulate the position of objects as well as handle occlusions if the object to be tracked is not visible from one of the sensors. On the other hand, an *inside-out* tracking system

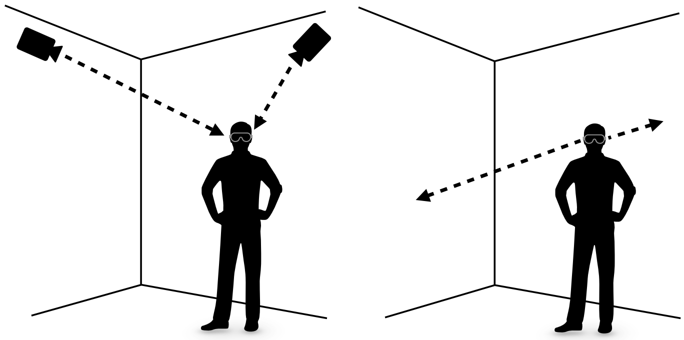


Figure 2.5: *Left*: In *outside-in* tracking systems, sensors placed in the environment track the position of objects. *Right*: In *inside-out* tracking systems, sensors on objects determine their position by observing the environment.

has the sensors on the object in the scene and the surrounding environment is being tracked (cf. Figure 2.5, right). Often the pose of the object with the sensor is calculated by *visual* tracking of objects in the environment.

Handheld AR often uses *visual* tracking.

Visual (or *optical*) tracking is often used in Handheld AR systems and "easily one of the most important physical tracking principles used today for AR" [Schmalstieg and Höllerer, 2016, p. 105]. Therefore, we discuss it in more detail. However, there are also many other tracking systems, from stationary systems using, for example, mechanical, electromagnetic, or ultrasonic tracking, to mobile systems with non-visual tracking such as GPS, magnetometers, or tracking with accelerometers and gyroscopes. All systems have different characteristics regarding accuracy, precision, and resolution and we refer to Schmalstieg and Höllerer [2016, pp. 96–104] for more information.

Within the area of visual tracking, there are numerous differentiations that affect the required capabilities of the system: *model-based* vs. *model-free*, *active illumination* vs. *passive illumination*, or *marker tracking* vs. *natural feature tracking*.

The distinction between *model-based* and *model-free* tracking refers to the type of prior knowledge that a system requires. For *model-based* tracking, the system looks for predefined information in the scene while for *model-free* tracking, the

Model-based: absolute position in a predefined model; *model-free*: position relative to the starting point.

model is build on-the-fly and does not require prior knowledge. While the *model-based* version allows the calculation of the absolute position in the predefined model, the *model-free* version estimates the pose only relative to the starting point of the tracking. In Handheld AR, both types of tracking are possible and we provide further information later in this chapter.

Active illumination adds light to the scene for tracking.

The different illumination types, *active* or *passive*, describe whether additional light is added to the scene, or whether only the existing light is used for tracking. *Active* illumination uses, for example, infrared light to detect reflective markers or determine depth information in the scene. In Handheld AR on current smartphones and tablets, *passive* illumination is the most commonly used technique. However, newer smartphones such as the iPhone 12 Pro[6] feature depth sensors to measure the distance between the device and objects in the scene.

Marker and *natural feature* tracking are both commonly used *model-based* tracking techniques.

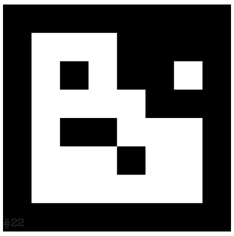


Figure 2.6: An example of an `arUco` marker.

Marker and *natural feature* tracking are both commonly used *model-based* tracking techniques. Both provide a predefined pattern that the AR system can look for in the environment. Based on the distortion of the pattern in the camera frame, the system can then calculate the position of the model relative to the camera. A *Marker* is often a black-and-white printout that has a previously specified pattern and orientation. An example for such markers are `arUco` markers [Garrido-Jurado et al., 2014] (cf. Figure 2.6). In an AR system using *active illumination*, reflective markers, for example on spheres, can also be used. In that case, multiple markers in a predefined arrangement are needed to differentiate between objects and calculate an unambiguous pose. Tracking with *natural features*, on the other hand, removes the need for a specialized marker. Instead, a real world object is scanned first and a virtual representation is used to find it in the camera view. This scan of a real world object can, for example, be a photograph. In the preparation to track the object, the AR system determines specific points in the photo—called *feature points*—to create a configuration pattern that it then looks for in the camera feed similar to *marker* tracking. With the feature points, the system can track the photo of the object like a marker. If the

[6] www.apple.com/iphone-12-pro/ (accessed 28.05.2021)

feature points are also stable from different vantage points and the real world object does not change greatly, it is also possible to track the object in the real world from the feature points generated from a photo. For details on the math involved in natural feature tracking, we refer to Schmalstieg and Höllerer [2016, pp. 138–149].

A *model-free* mobile Augmented Reality system can use algorithms such as *SLAM* to track its view transform at runtime. *SLAM* stands for *Simultaneous Localization and Mapping* and describes a process in which feature points are detected in the camera frame and tracked incrementally over time [Schmalstieg and Höllerer, 2016, pp. 156–164]. This allows the system to build up a model while using the AR system and track the pose of the camera relative to elements in the scene.

Model-free mobile AR systems detect and track feature points during use.

In many advanced AR devices, such as Microsoft's HoloLens or Handheld AR on current smartphones, tracking is not only relying on one method alone. Instead, multiple tracking methods are used together to improve overall tracking quality. For example, tracking in current Handheld AR devices combines visual tracking with data from other sensors, such as accelerometers and gyroscopes, to determine their pose. This process of combining calculations from different sensors is often called *sensor fusion* [Schmalstieg and Höllerer, 2016, pp. 117–120].

Multiple tracking methods can be used together to improve tracking quality.

2.3 Depth Perception in AR

Augmented Reality—and in particular Handheld Augmented Reality—has several perceptual issues [Drascic and Milgram, 1996; Kruijff et al., 2010]. A big reason for these issues is that humans are so used to and sensitive about the perception of the real world, that slight deviations affect the perceived realism in virtual environments [Schmalstieg and Höllerer, 2016]. These deviations can, for example, be caused by less than perfect tracking, latencies when displaying the virtual objects, or conflicting depth cues.

AR has several perceptual issues.

Users have issues detecting the correct depth of objects.

In a survey of usability studies in Augmented Reality by Dey et al. [2018], the area of *perception* was the second highest studied area after *interaction*. An often reported issue, not only for Augmented Reality but also for Virtual Reality, is that users have issues detecting the correct depth position for objects and frequently underestimate distances [e.g., Armbrüster et al., 2008; Diaz et al., 2017; Wann et al., 1995]. Several of these studies not only focus on investigating visual perception issues but also on how to address them for Augmented Reality systems. Dey et al. [2018], Dey and Sandor [2014], and Swan et al. [2017] published surveys on this topic and we provide further details on approaches when we evaluate depth visualization techniques in Chapter 8. In this chapter, we describe the components of depth detection and how they affect (Handheld) AR.

Depth cues help perceiving the spatial order of objects.

Depth cues are clues that humans perceive and interpret to understand the spatial arrangement of elements in the scene [Cutting and Vishton, 1995]. In virtual environments, these cues need to be reproduced to create a realistically looking scene [Schmalstieg and Höllerer, 2016; Cipiloglu et al., 2010; Drascic and Milgram, 1996]. Cutting and Vishton [1995] mention three categories of *depth cues*: *physiological*, *kinetic*, and *pictorial*.

Physiological cues are perceived by interpreting differences in the images of the eyes.

Physiological cues describe the perception of depth based on differences between the images that each eye perceives. Several studies have evaluated this form of depth perception in stereoscopic systems [e.g., Ellis and Menges, 1998; Jurgens et al., 2006; Livingston et al., 2009]. Handheld AR, however, does not allow for this type of depth cue in most cases as the camera image and the display cannot provide stereoscopic depth cues of the real world scene. Instead, the stereoscopic vision of the user receives cues about how the entire handheld display itself is positioned in space.

Kinetic depth cues are based on movement.

Kinetic depth cues are based on movement of the viewer (or viewing device) and/or movement of objects through the environment. *Motion parallax* is one example of kinetic depth cues and it refers to the perception of different movement speeds and directions depending on the object's distance to the viewer [Cutting and Vishton, 1995; Drascic and Milgram, 1996]. An example for this is the perception when riding a train that mountains in the distance move slower than the trees on the side of the railroad tracks. In Handheld AR, moving the device around can help in using this kinetic depth cue and it requires less movement compared to a head-mounted device. However, it is also one reason for the *dual view* issue mentioned by Kruijff et al. [2010] that affects the perception of the scene because (real) objects in the scene are seen twice, once on the screen of the Handheld AR device and once directly by the user.

Pictorial depth cues are known from painting and photography.

Pictorial depth cues are known from traditional painting and photography. Cutting [2003] mentions five pictorial depth cues: *occlusion*, *relative size*, *relative density*, *height in the visual field*, and *aerial perspective*. Their impact, however, is also affected on the distance to the viewer. Here, Cutting splits this distance into the categories *near* (or *personal*) space, *middle* (or *action*) space, and *far* (or *vista*) space. For our work in this thesis, the near space enclosing the area within arm's reach of the viewer, is most relevant. In the near space, *occlusion*, *relative size*, and *relative density* are helpful pictorial depth cues [Cutting, 2003]. Current Handheld AR systems use graphics frameworks that include many of these depth cues automatically so that aspects such as relative size or occlusion between virtual objects look "right". While the occlusion of real objects by virtual objects works well especially in Handheld AR systems, since the rendering of the virtual object on top of the camera stream occludes the real object behind it, occlusion of virtual objects by real objects is more problematic. For this occlusion to work, the geometry of the physical object needs to be tracked so that a virtual *phantom object* can be added to the scene to correctly occlude the virtual object [Schmalstieg and Höllerer, 2016, pp. 199–200]. However, tracking the geometry of physical objects is difficult, particularly in model-free systems as not only the position but also the whole geometry needs to be determined at runtime. Schmalstieg and Höllerer [2016, pp. 202–205] provide an overview of model-free occlusion techniques, but so far these are not supported by popular Handheld AR frameworks. Current research projects aim to make the use of depth information simpler for handheld displays [Du et al., 2020] or use a neural network to blend real and virtual objects [Tang et al., 2020]. The recent addition of depth sen-

Occlusion of virtual objects by real objects is difficult in Augmented Reality.

sors, such as the LiDAR sensor in Apple's iPhone 12 Pro models, could make correct occlusion by physical objects available further for a wide audience.

X-ray applications enable seeing through objects.

An area in which adapting occlusion is particularly relevant, is the area of X-ray applications that enable users to see objects that are hidden behind other objects [e.g., Dey et al., 2012; Dey and Sandor, 2014; Tsuda et al., 2005]. These applications are, for example, used in a navigation context to show buildings or structures behind other buildings. To show areas behind physical buildings, the visualizations in X-ray applications either remove an area of the occluding structure completely, or keep elements of the foreground in view to preserve context [Dey and Sandor, 2014]. The focus of studies on X-ray visualizations is mainly to show the existence of objects behind others than precise distances between them.

Different fields of view cause perceptual issues in Handheld AR.

Another issue especially relevant for Handheld AR is caused by conflicting fields of view. This is another part of the *dual view* issue mentioned by Kruijff et al. [2010]. The camera in a Handheld AR device often has a wide angle lens that captures more of the scene than what would normally be visible if the device was just a plane of glass. This is further enhanced by the placement of the camera which is mostly not centered on the back of the device. The result of the placement and viewing angle is that users can see parts of the environment on the device while simultaneously also seeing them when looking past the device. Research projects aim to reduce this effect by using *user perspective rendering* which tracks the head position of the user looking through the device to adapt the portion of the camera view that is shown on the screen [Schmalstieg and Höllerer, 2016, p. 70] [Čopič Pucihar et al., 2013; Gombač et al., 2016]. However, while several smartphones are capable of tracking the head of the user through their front-facing camera, user perspective rendering is not yet available in the main frameworks for Handheld AR.

Having established the main concepts and issues of Augmented Reality in this chapter, we look at the impact of being able to interact on real physical objects compared to virtual objects, in the next chapter.

Chapter 3

Physical Guides

In this chapter, we investigate the effect of sketching on physical objects compared to virtual objects and how different surface properties impact the performance. Compared to Virtual Reality, Augmented Reality has a closer relationship with the real, physical world [Milgram and Kishino, 1994] and enables sketching directly on physical objects in this space. The created sketches could either extend the existing object or be used to create a new, matching object. An example is the inset for the cup holder we mentioned in the introduction or a new lampshade to replace a broken one. Enabling users to achieve tasks like this in a simple way without requiring extensive knowledge of professional modeling tools, makes 3D object design more approachable for novices. To design an object dependent on an existing object in Virtual Reality would require the user to digitize the existing model first to be able to see it in the virtual environment. In Augmented Reality, however, she can use the existing physical object directly instead, such as the existing cup holder and the can that should fit into it.

We investigate the effect of physical objects on sketching performance in AR.

Publications: The work presented in this chapter has been done in collaboration with Adrian Wagner, Simon Voelker, and Jan Borchers. It has been published as a poster at ACM CHI '18 [Wacker et al., 2018a], as a full paper at ACM SUI '18 [Wacker et al., 2018b], and in the master's thesis of Adrian Wagner [A. Wagner, 2018]. The author of this dissertation is the main author of the conference publications and developed the research idea and motivation. Most sections in this chapter are taken from the full paper publication. The study data has been reanalyzed for this thesis.

Additionally, using physical objects allows using the haptic feedback their surface provides for guiding the stroke in the modeling task. For such a task, it is helpful to understand how well users can draw along a planned line on a physical object.

Studies in VR have shown that a flat physical surface improves stroke accuracy.

Previous work in VR has shown that sketching planar shapes is more accurate when they are drawn on a physical surface compared to drawn in mid-air [Arora et al., 2017]. Real objects, for example a water bottle, however, have a more complex shape and structure and offer a variety of guidance elements such as visual guides like a printed line or haptic guides given by curves and edges of the object.

In this chapter, we explore the space of designing objects that match existing objects in an AR environment. For this, we first look at related work from the areas of *modeling supports*, *haptics in VR*, and how researchers are using *physical objects as guides* (Chapter 3.1). Then, we present a classification of the various guidance types that physical objects offer (Chapter 3.2), and study their impact on the 3D sketching performance in Augmented Reality (Chapter 3.3). As designing models can also be done around virtual objects placed in the environment, such as a desk model when planning a new office [IKEA, 2017], our classification and study also includes virtual representations of each guidance type. After discussing our findings, study limitations, and additional insights (Chapter 3.4), we end this chapter with suggestions for future work in this area (Chapter 3.5).

In summary, we make the following contributions in this chapter:

- a classification of guidance types when drawing on existing objects in AR,
- the first lab study to quantify the impact of different guidance types on time and accuracy when sketching on non-planar physical and virtual models in AR.

While our classification can be used by researchers to structure experimental conditions and describe object features more consistently, our study findings present initial metrics for researchers and designers of AR sketching systems to assess the effect of different guidance types on the precision to expect for a drawing operation.

3.1 Related Work

Besides being influenced by related work from 3D modeling areas (cf. Introduction in Chapter 1), the work in this chapter is influenced by three other areas: *modeling supports*, assisting in the creation of sketches and objects, *haptics in VR*, adding haptic feedback to virtual environments, and using *physical objects as guides* that help when creating digital models.

3.1.1 Modeling Supports

Several projects have investigated systems that aim to improve model generation.

To improve the quality of models created, researchers have proposed a variety of guidance systems. The Virtual Reality Multiplanes system by Machuca et al. [2018] displays snapping targets that the user can select to, for example, create a perfect right angle. This enables users to create objects with straighter lines than possible with freehand drawing. WireDraw by Yue et al. [2017] supports creating a physical wireframe of a known 3D model with a 3D extruder pen, by displaying stroke guides in AR over the emerging physical object. PapARt [Laviole and Hachet, 2012] lets users manipulate a 2D projection of a 3D scene, then simplifies it for easy tracing by the user. Similarly, Flagg and Rehg [2006] decompose an artwork into layers that are projected onto a canvas sequentially to help the user replicate it. Other systems help aligning strokes by displaying visual guides [Iarussi et al., 2013] or adjusting the stroke afterward [Fernquist et al., 2011]. Rivers et al. [2012] project guides onto a solid raw sculpture to indicate where to add or remove material to reach a predefined shape.

Based on studies of how well humans can follow paths [Pastel, 2006], Cao and Zhai [2007] provide a model of human pen strokes that reveals that, while humans slow down at stroke corners, the exact angle of the corner hardly impacts speed. While performance of freehand sketches improves with practice [Wiese et al., 2010] and users can recreate patterns and positions without visual feedback [Gustafson et al., 2010], the missing haptic feedback still affects their precision.

Freehand sketches improve with practice but missing haptic feedback affects precision.

3.1.2 Haptics in Virtual 3D Environments

Missing haptic feedback is a well-known major limitation of VR [Brooks, 1999; McNeely, 1993]. For VR, Arora et al. [2017] showed that displaying visual guidelines, such as a surface and/or optimal path, already improves the drawing accuracy (overall and projected deviation from the optimal stroke) of freehand strokes, while providing a physical surface as guide improves it dramatically. The physical surface used in their study was a flat board and the target circle was aligned to this board.

Visual guidelines improve drawing accuracy but less than a physical surface.

Fleisch et al. [2004] use various input devices for sketching on a semi-immersive virtual table, with physical 'input planes' to align a digital plane with the real table surface as an aid in drawing strokes. Mockup Builder by De Araùjo et al. [2012] combines the precision of using a touchscreen with the realistic viewpoint manipulation of a VR headset for gesture-based modeling. Similarly, Arora et al. [2018] also combined immersive sketching with a physical drawing surface. Users can sketch in Augmented Reality as well as define drawing planes. Subsequent sketches on the physical tablet are then projected onto the drawing plane. The users benefitted from the expressiveness of free mid-air sketching while also having the precision of a physical drawing surface available. Drey et al. [2020] and Surale et al. [2019] explore how a tablet can be used in Virtual Reality environments, e.g., for sketching and modeling tasks. They provide analyses of the design space of this interaction with an overview of existing research projects. Peng et al. [2018] developed a system that allowed the user to create and manipulate 3D geometry in Augmented Reality while a robot simultaneously prints the physical model in

Research projects have used the benefit of haptic surfaces by aligning the virtual content with them or projecting input on them back into the scene.

the same space. This enables users to use the created physical wireframe during the modeling process.

Passive and active haptic feedback increase immersion in virtual systems.

The feeling of immersion is also affected by missing haptic feedback in virtual systems [Insko, 2001]. Even passive haptic elements already improve immersion in VR, e.g., a ridge on the floor improves the feeling of standing at the edge of a cliff [Insko, 2001]. Active haptic feedback increases both immersion and accuracy of input in VR: Creating haptic constraints improves the capability to draw 3D curves [Keefe et al., 2007]. Many research projects investigate ways to simulate touching virtual objects, for example, by stimulating arm muscles electrically [Lopes et al., 2017] or through wires attached to the fingers and hand [Fang et al., 2020]. Other setups aim to bring haptic props, such as a fishing rod, to the location of interaction, for example, by having a circular array of props rotating around a platform the user is standing on [Huang et al., 2020]. Finally, static physical objects can assist input in VR. Performing gestures in relation to physical printouts can simplify input to a VR system [Jackson and Keefe, 2011], even though this study did not provide in-place visual feedback of the stroke. 3D printouts of corals, for example, have been used to navigate data about them in VR [Kruszyński and Liere, 2009], and physical maze elements have assisted novices creating VR mazes [Gai et al., 2017].

Static physical objects can assist input in VR.

3.1.3 Physical Objects as Guides in Personal Fabrication

Personal Fabrication often requires aligning virtual and physical objects, for example when designing an object to 3D-print that should fit around or inside an existing object.

Systems use physical objects during the printing process or to specify input while modeling.

K. Zhu et al. [2016] use physical objects such as pens during the 3D printing process to create exact cutouts on printed objects. MixFab by Weichel et al. [2014] lets users place small physical objects behind a see-through display and create virtual models aligned to them, e.g., to cut holes in the virtual object that fit the physical object. Weichel et al. [2015] also present physical measurement tools, such as a caliper, that communicate with a digital development tool to send physical measurements to the computer or digital values back to the physical tool. In ModelCraft by Song et al. [2006], users sketch directly on folded paper objects covered with a printed marker pattern that lets a special pen send each sketched edit and annotation back to the linked digital model.

In summary, while research has been exploring sketching in AR from several directions, there has been no quantitative study of how different types of physical and virtual guidance affect the precision we can expect when users sketch in Augmented Reality.

3.2 Classification of Guidance Types

Guidance types are features that assist drawing a stroke.

To structure our investigation, we first introduce a classification of guidance types. These guidance types are elements that objects may offer to a user tracing shapes on their surface with a pen to create strokes that are aligned to the object. While many factors affect drawing on physical objects, from material properties such as hard, soft, rough or smooth surfaces, to object size, location, and orientation, we focus on local object features that may guide a stroke.

A surface limits 3D movement to a plane

In the simplest guided case, the user is drawing on a flat surface, such as a table (cf. Figure 3.1). If we consider this surface as the xy-plane, then this restricts movement in the $-z$ direction. This is a hard constraint since the user cannot press into the table surface. It reduces pen movement by half a degree of freedom. To avoid lifting the pen off the surface in the $+z$ direction, the user exerts some pressure while tracing. This removes another half degree of freedom. However, this is only a soft constraint, since the user can still move in that direction. Therefore, the guidance to draw on a flat surface can be seen as an even mix of hard and soft constraints.

Edges restrict movement to a line.

Nonplanar surfaces provide additional tracing guides to the user, such as the edge of a table. Such guides aim to reduce the movement to a line, and enable the user to trace

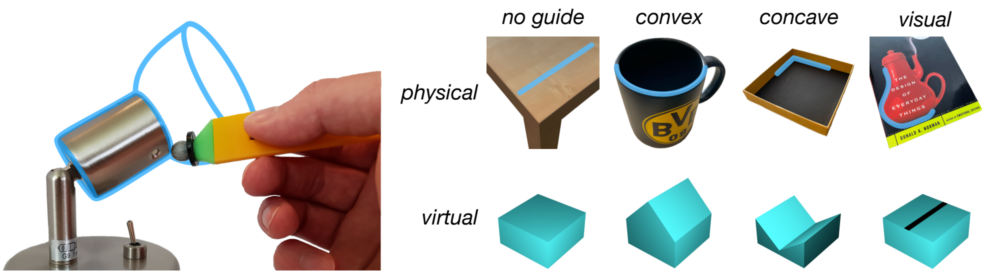


Figure 3.1: Augmented Reality lets users sketch directly on the surface of a physical object to design a new model that fits parts of the existing object (*left*). We classified the different surface guides that physical and virtual objects provide (*right*).

particular features of the object. While none of these non-planar surface features completely remove another degree of freedom (this would require locking the pen tip into a surface rail), how hard or soft their constraint is depends on the physical shape of the guide. A *concave* form along the line the user wants to trace, e.g., around the curved neck of a vase, is a harder constraint since the user's pen pressure will push the pen tip towards the guide. A surface that is *convex* along the line the user is tracing, such as the opening of the vase, is a softer constraint since the user can easily slip off the ideal line. The more concave or convex the surface, the stronger the constraint. The extreme cases of these guides are especially common and worth studying: concave edges, as when tracing the inside edges of a box, and convex edges, like the outside edges of a box, or the table edge mentioned above.

Of course, visual markings on a surface can also guide the user in drawing a particular line. They can be natural, like the grain in a wooden table or the water line inside a bottle, or artificial, like a printed line on a book cover. We refer to these guides as *visual* guides. They are soft constraints, because they remove a degree of freedom when tracing them, but without providing any physical, haptic guidance.

Visual elements can also reduce movement to a line.

Therefore, tracing a particular feature of a physical object can be understood as a limitation of two degrees of freedom involving both hard, physical and soft, "logical" constraints [Norman, 2013]. The first limited degree constrains movement to the surface of the object, as with a free-hand

Tracing a feature of an object is affected by hard and soft constraints.

stroke drawn on a table. At this stage of the classification, there is no difference between drawing on a flat surface like a table or around a cylindrical object like a bottle. In both cases sketching is limited to the surface by equal measures of hard (cannot press into the object) and soft (should not lift pen off the surface) constraints. A second limited degree of freedom constrains movement to along a one-dimensional line, straight or curved, on the object, using its physical shape or surface markings.

Virtual objects in AR feature the same guidance types, only with soft constraints.

So far, we have only considered drawing on physical objects in AR. However, since AR supports showing virtual objects in the real world, tracing them should also be considered for comparison. For example, when planning a kitchen in AR, a user may want to trace on a virtually displayed working surface to outline the cutout for the sink. In this case, the first degree of freedom that constrains movement to a surface is already a soft constraint, because it is physically possible to penetrate the surface of a virtual object. Similarly, convex and concave features on virtual objects are only soft constraints, as are surface markings.

Surface guidance limits movement to a plane, *line guidance* limits it to a line.

In conclusion, the guidance types on an object can be seen as limiting the degrees of freedom for sketching. The first limitation guides the free-hand movement to a surface by constraining one degree of freedom (*surface guidance*). This guidance can be either *physical* or *virtual*. On physical objects, this constraint is hard for one half degree of freedom and soft for the other half. On digital objects, it is entirely a soft constraint. Movement can be limited further to tracing a line or curve by *visual* guides and object shape (*line guidance*). *Concave* shapes provide more guidance, reducing the degree of freedom more than *convex* shapes, and curvature also increases guidance, with *concave* and *convex* edges as extremes. This results in eight combinations (cf. Table 3.1).

Constraint	*Virtual*	*Physical*
No guide	Virtual table surface	Real table surface
Visual	Pen stroke on a virtual sketch	Waterline in a bottle
Concave	Inside of a virtual bucket	Intersection of shelf & wall
Convex	Edge of a virtual desk	Opening of a wine glass

Table 3.1: Examples for all combinations of initial constraint & subsequent limitation to a line.

3.3 Tracing Study

Our study was designed to quantify the effect of different guidance types on tracing time and accuracy. The results can help AR sketching systems to filter raw pen movements based on the context to determine intended strokes. Knowing when users want to move to mid-air and not follow a certain guidance on the object anymore, could be used to switch between smoothing the performed stroke to "snap to the real object" and recording the mid-air sketch.

We studied the effect of guidance types on stroke performance.

To focus on this influence, we chose the basic task of drawing on the surfaces of fixed, upright objects and kept other properties such as size consistent. The size of our objects matched things typically created in the small build volumes of affordable 3D printers. We chose a cylinder and a cuboid as our basic object types, around which the user had to draw a *circle* and *square* respectively. This way we covered both continuous and non-continuous motions. We decided to use fixed stationary objects to avoid effects of varying grips and of rotating the object instead of moving the pen to draw. It also let us include *virtual* objects without adding confounding variables such as handling and moving of digital objects. We measured accuracy by comparing the trace to the optimal shape.

Users had to draw around fixed, upright objects.

We compare the stroke to the optimal shape.

3.3.1 Experimental Design

For each object (cylinder/*circle* and cuboid/*square*), we included both *physical* and *virtual* models as conditions to reflect the *surface guidance* described before. Our *line guidance* types describe four conditions limiting movement to a line: *no guide* as baseline, *visual*, *concave*, and *convex*. For this study, we focused on edges as the most pronounced and common *convex* and *concave* surface features.

We included all guidance types in the study.

The result is a 2×2×4 design: *shape* (*circle* / *square*) × *surface guidance* (*physical* / *virtual*) × *line guidance* (*no guide* / *visual* / *concave* / *convex*).

Participants performed 5 strokes for each condition.

The sequence of independent variables was counterbalanced with a Latin square. Participants performed 80 strokes total, five strokes in each condition. They could practice each condition before their five strokes. A session took 45–60 min.

3.3.2 Participants

We collected data with 16 participants.

Two of our initial 16 participants expressed general issues with detecting the position of objects displayed by our AR headset. We excluded their results from the evaluation, and recruited two replacements to fill the Latin square again. Of our final 16 participants (4 female, 19–29 years, M = 24.9 years, SD = 2.4 years, all able-bodied), 9 wore glasses but were near-sighted at a level sufficiently low to use our headset without glasses. 11 had no prior experience with AR in general or VR drawing tools; 4 had experience with AR; 1 had no AR experience but had used a VR drawing tool in the past; no one had experience with both.

3.3.3 Apparatus

We used 3D-printed objects featuring the guidance types.

Our 3D-printed physical objects for tracing were 16 cm high, 8 cm wide, and could be mounted normally or upside down. A different *line guidance* was placed at 4 cm from each end. This let us combine *no guide* and *visual* on each green object, and *convex* and *concave* on each red object (cf. Figure 3.2, back). The object was mounted on a plate in front of the participant. The plate was attached to the table to avoid accidental movements.

Pen Design & Tracking

We tracked a custom pen with a motion tracking system.

We created a custom pen (cf. Figure 3.2, front) tracked by a Vicon[1] motion tracking system that used six high-speed infrared cameras to track reflective markers from different an-

[1] www.vicon.com/ (accessed 28.05.2021)

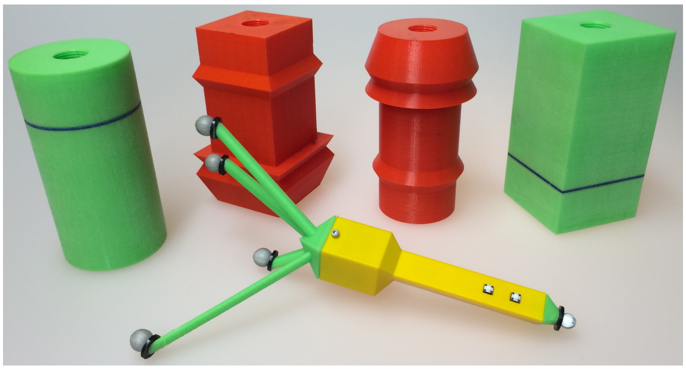


Figure 3.2: *Front*: Custom pen used in the study. *Back*: Physical objects used in the study. For each *shape* condition, we had two physical objects to reflect the *no guide* and *visual* condition as well as the *convex* and *concave* condition.

gles at 100 fps with sub-millimeter accuracy. We chose the shape of pen for the input device, since it is a well known form factor for drawing tasks and it can be held in a *precision grip*, often used for fine input [LaViola et al., 2017, p. 70]. Similar to other projects [e.g., Jackson and Keefe, 2016], our pen featured markers at the end and tip. Using a spherical marker as the tip both improved tip tracking stability and prevented user confusion when the alignment of the virtual line rendering drifted slightly around the center of the physical tip. The pen included two buttons for inking and calibration, and a Bluetooth LE module to send their states to a receiver. A Mid-2012 MacBook Pro running our user study software extracted the pen tip position in 3D space from the Vicon data. While the Inking button was pressed, the software recorded the pen tip position as a path, and forwarded it to a Microsoft HoloLens headset to render the path into the user's view. We decided to use the *outside-in*, *active illumination* tracking because the HoloLens did not allow for precise, continuous tracking of the *model transform* of a pen.

The pen's position and button information were sent to an HMD for visualization.

Visualization (Registration & Rendering)

Using a Microsoft HoloLens headset allowed us to visualize the drawn stroke and the *virtual* objects with high sta-

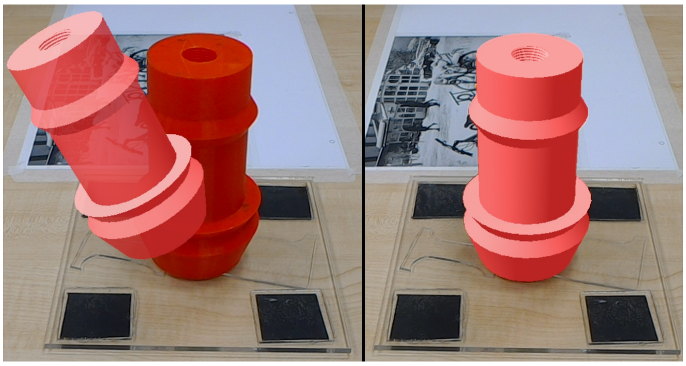


Figure 3.3: After detecting the visual marker, Vicon and HoloLens coordinate systems are not perfectly aligned yet (*left*, exaggerated). We manually adjust them so that the virtual model is aligned precisely with the physical model (*right*).

The coordinate systems of the HMD and tracking system needed to be aligned.

bility. This required aligning the Vicon and HoloLens coordinate systems. Our calibration setup used the Vicon calibration wand and a visual marker that the HoloLens could track. Knowing their positions in reality, we could roughly align both coordinate systems. To address limitations of the visual tracking of the HoloLens, we then fine-aligned the coordinate systems manually by adjusting rotation and location of a digital model to fit a real-world counterpart placed at a known point (cf. Figure 3.3). This allowed us to calculate where a point measured by the Vicon would need to be displayed in the HoloLens, to render a sphere on top of the physical pen tip as it moved around the room, and to render strokes in place while inking (cf. Figure 3.4). While the user pressed the Inking button, we rendered a line with 30 fps using the most recent points forwarded to the HoloLens, to visualize the path the user was drawing. We occluded those parts of the path behind the model in both the physical and the virtual condition to preserve realism, based on pilot tests (cf. Figure 3.4 top, left).

3.3.4 Study Procedure

Participants sat at a table inside the Vicon's tracking volume (cf. Figure 3.4, top, right). They were allowed to move

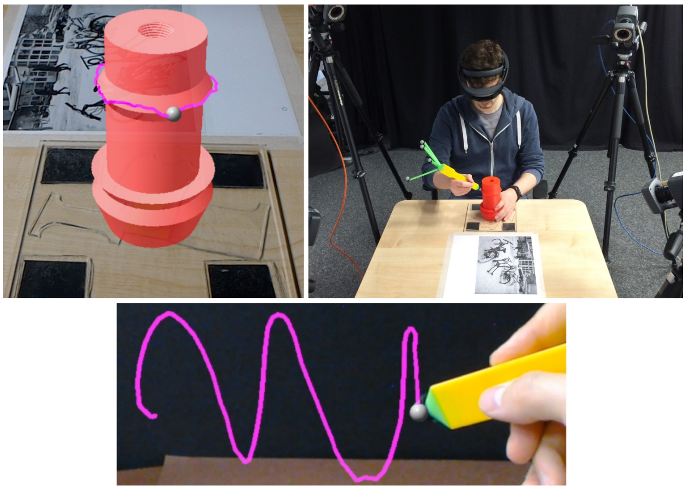


Figure 3.4: After aligning the coordinate systems, points reported by the Vicon can be rendered into the viewing area of the HoloLens (*bottom*). In our study, participants were asked to draw a stroke around virtual objects (*top*, *left*) and physical objects (*top*, *right*).

their head and torso but were asked to remain seated. Each trial started by showing the object to trace around. In *physical* conditions, we screwed the object onto the mounting plate, and asked participants to grab the object with their non-dominant hand while drawing. In *virtual* conditions, we asked participants to rest their non-dominant hand on the mounting plate. The participant was allowed to move their hand and the pen through the *virtual* object. We asked our participants to draw around the object with a regular drawing speed, while keeping precision in mind.

The objects were placed in front of the seated participant.

During the implementation of the system, we observed that the HoloLens occasionally adjusts its coordinate system due to updated tracking information from the environment. If this happened after calibration of our system, the coordinate systems of Vicon and HoloLens became misaligned, making the real and rendered pen tip deviate from each other by up to 10 mm. We asked participants to mention any offset to us during the study and also inquired about the correct alignment occasionally throughout the session. In case of a misalignment, we re-synchronized the coordinate systems before continuing. This calibration was necessary once each participant first mounted the

Misalignment due to updated tracking information required recalibration of the system.

HoloLens, because a user's individual physiology affects the alignment of the HoloLens. We encountered the aforementioned need to recalibrate during sessions only twice, and re-ran the last trial after recalibration.

3.3.5 Measurements

We compared the processed stroke to the optimal path.

Similar to the study from Arora et al. [2017], we processed the data for each stroke with a low-pass filter averaging over a 10 frame window, created a path from the resulting points, and then resampled the path to 100 equidistant points for evaluation. This removed a potential bias due to the higher point count in areas such as corners where participants slowed down. We compared these resampled points to the optimal stroke to calculate five measurements:

Mean Deviation in 3D

3D deviation describes the overall deviation.

We calculated the shortest distance from each point to the target shape and calculated the mean (*3D deviation*). This represents the mean deviation from the target shape similar to the calculation by Arora et al. [2017].

Mean Deviation in x&y Direction

x&y deviation measures the effect of *surface guidance*.

We projected the sampled points onto the surface plane of the table, and computed the mean difference from the projected target shape (*x&y deviation*). This measured the effect of *surface guidance* (*physical* or *virtual*). In the *physical* conditions, this corresponded to 'lift-offs'.

Deviation in z Direction

z deviation measures the effect of *line guidance*.

For this, we evaluated only the z coordinate of each sampled point and calculated the deviation from the target height. For the *no guide* condition, we set the height of the

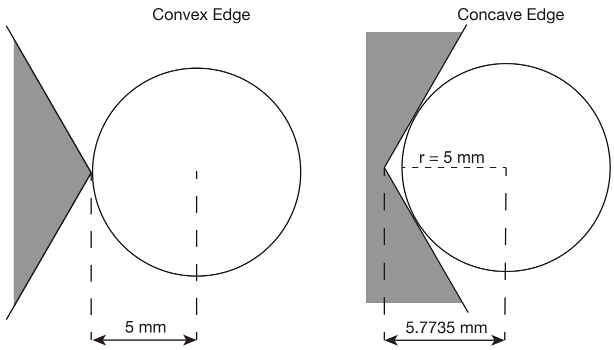


Figure 3.5: Optimizations made to the target stroke for the *physical* conditions to correct the inability to draw directly on the surface.

target stroke to the height of the first recorded drawing position for each stroke. This deviation evaluates the effect of *line guidance*, as it measures the deviation in the dimension unconstrained by *surface guidance*. We split this value up into two sub-classes:

1. The *average absolute deviation* from the target height (*z deviation*).
2. The *average directed deviation* from the target height (*directed z deviation*). This allows us to evaluate whether a stroke was mainly above or below the target height.

Since the line drawn originated from the center of the spherical marker at the pen tip, the marker displaced the user's input on physical surfaces by its radius of 5 mm, making it impossible to perfectly trace physical surfaces. To account for this, we virtually enlarged all physical target shapes by that radius for our calculations. The *physical*, *concave* condition required another enlargement to a total of 5.77 mm since the radius of the pen tip marker was larger than the concave opening, keeping it further away from the concave edge (cf. Figure 3.5). Adjustments were not necessary for the *virtual* conditions since the user could penetrate object surfaces to align strokes.

We enlarged the optimal path for physical objects.

Stroke Duration

duration measures the drawing time.

We also recorded drawing time by measuring the elapsed time between the first and last inking operation for each stroke (*duration*).

3.3.6 Results

We analyze the data using confidence intervals.

Due to measurement issues, we discarded 20 strokes of the 1280 recorded strokes. For every participant, we averaged the dependent variables over the five trials to get one measurement for each guidance type and participant. We analyze our results by using confidence intervals and estimation. We provide reasonings and information regarding this evaluation style in Appendix A. The results found using this evaluation show the same major findings as the evaluation in the published paper [Wacker et al., 2018b]. However, the visual representation of data makes the effect sizes more apparent in the evaluation in this thesis.

We show the overall confidence intervals without within-subject adjustment in Figure 3.6.

We look at within-subject differences.

In the following sections, we look at within-subjects mean differences regarding our measurements to estimate the effect that using a particular guidance type has compared to a baseline. For this, we subtracted the averaged value of the baseline from each of the averaged values of the other guidance types. This means for the following comparison plots that a value to the right of zero indicates more, for example, *3D deviation* compared to the baseline. For *surface guidance*, we take *physical* as our baseline and for *line guidance*, we take *no guide*.

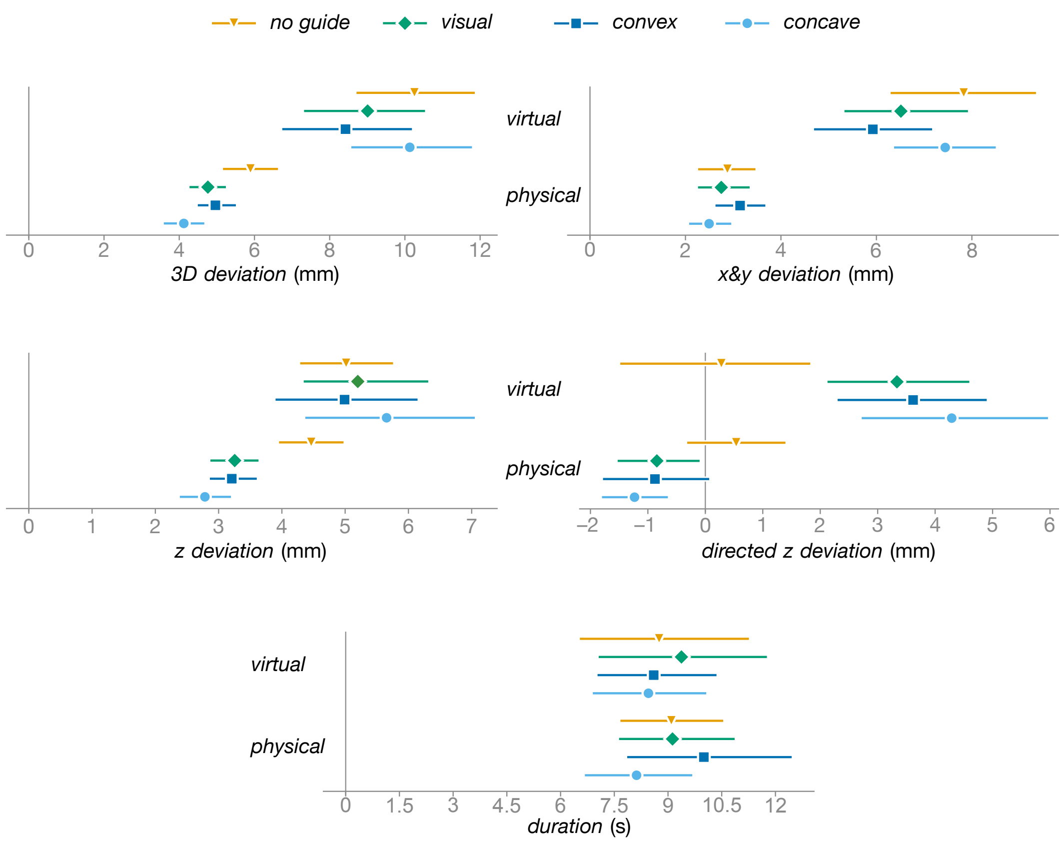
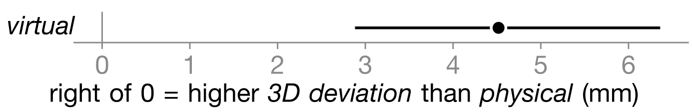


Figure 3.6: Results of the tracing study. Means and 95 % CIs without within-subject adjustment.

Mean Deviation in 3D

The plot below shows the main comparison of the *3D deviation* between *physical* and *virtual* objects. As the values are clearly to the right side of '0', this shows that measurements on *virtual* deviate more compared to *physical* (*physical*: M = 4.9 mm, CI [4.5 mm, 5.4 mm]).

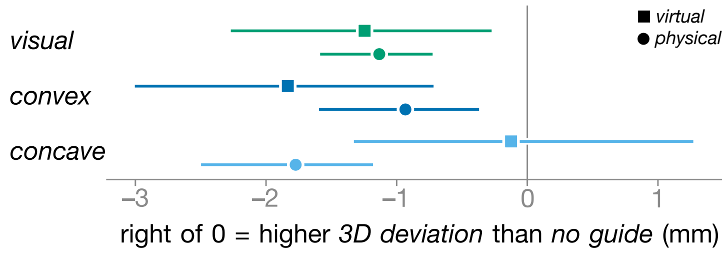
Strokes deviate more on *virtual* objects.



Looking at *line guidance*, we see that all combinations except *virtual*, *concave* are to the left of '0' meaning that having *no guide* results in more *3D deviation* (*physical*, *no guide*:

Line guidance mostly improves accuracy.

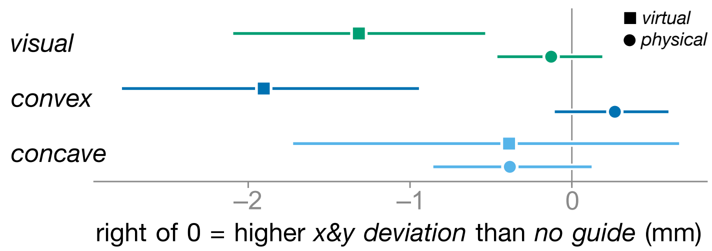
M = 5.9 mm, CI [5.2 mm, 6.6 mm]; *virtual, no guide*: M = 10.3 mm, CI [8.7 mm, 11.9 mm]). *Virtual, concave* seems to have similar *3D deviation* compared to the baseline. Among the other guidance types for each *surface guidance*, there seem to be no great differences with a slight trend that *physical, concave* and *virtual, convex* have less *3D deviation*.



Mean Deviation in $x\&y$ Direction

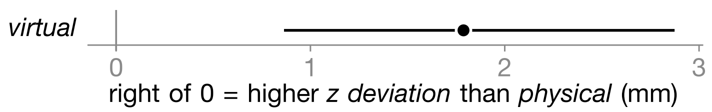
The effect of *line guidance* on physical objects is reduced for $x\&y$ deviation.

The overall results for *x&y deviation* show similar trends as the *3D deviation*, especially for the difference between *physical* and *virtual* objects. For *line guidance*, the measurements for *physical* move closer to the values of the baseline (*physical, no guide*: M = 2.9 mm, CI [2.3 mm, 3.5 mm]; *virtual, no guide*: M = 7.8 mm, CI [6.3 mm, 9.3 mm]).



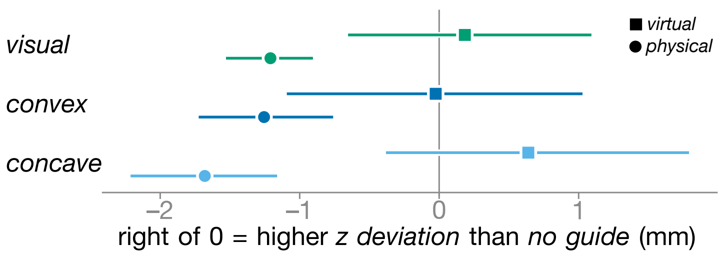
Deviation in z Direction

Looking at the *z deviation*, the difference between *physical* and *virtual* becomes smaller but there is still more deviation for *virtual* than for *physical* objects (*physical*: M = 3.4 mm, CI [3.1 mm, 3.8 mm]).



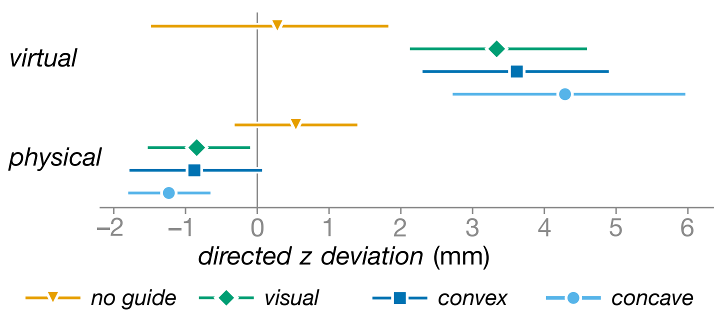
Regarding the *line guidance*, the guidance type does not seem to affect the *z deviation* for *virtual* objects (*virtual*, *no guide*: M = 5.0 mm, CI [4.3 mm, 5.8 mm]). However, on *physical* objects, having any form of guidance appears to reduce the deviation in *z* direction (*physical*, *no guide*: M = 4.5 mm, CI [4.0 mm, 5.0 mm]).

Line guidance does not effect *z deviation* on *virtual* objects.



For the *directed z deviation*, we compare the height of the lines drawn compared to the target height. Therefore, we evaluate the overall measurements (also shown in Figure 3.6) instead of differences to our baselines. Comparing these heights indicates that lines on *virtual* objects with any form of *line guidance* are above the target line. On the other hand, lines on *physical* objects with any *line guidance* seem to be drawn slightly below the target line. Both *no guide* conditions are drawn close to the target line with a slight tendency to be drawn above the target height.

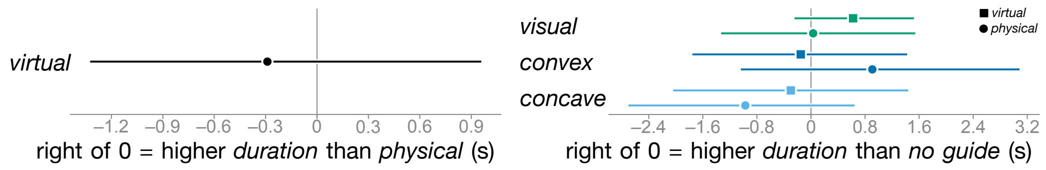
Participants drew above the target line on *virtual* objects.



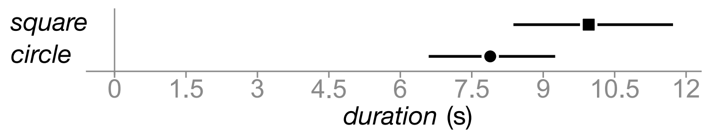
Stroke Duration

Guidance types have no effect on *duration*.

Regarding the *duration*, our results do not indicate great differences regarding both *surface guidance* and *line guidance* (*physical*, *no guide*: M = 9.1 s, CI [7.7 s, 10.5 s]; *virtual*, *no guide*: M = 8.8 s, CI [6.5 s, 11.3 s]).



However, this is the only instance in which *shape* seems to have an effect as participants drew faster around a *circle* than around a *square*.



3.3.7 Front-to-Back Comparison

Stroke renderings suggest differences between the front- and back-facing sides.

Studying aggregate renderings of the strokes performed, as in Figure 3.7, we noticed a pattern that motivated us to compare accuracy on the front- and back-facing halves of each object, to evaluate how well users could continue strokes they could no longer see.

We split recorded points into a front-facing and a back-facing half. We computed the mean *3D deviation*, mean *x&y deviation*, and mean absolute *z deviation* for both halves. We re-did the evaluation and report all results related to the new variable *side*. For the within-subjects differences, we calculated the difference between the *back* and *front* so that values to the right of '0' indicate more deviation on the *back*.

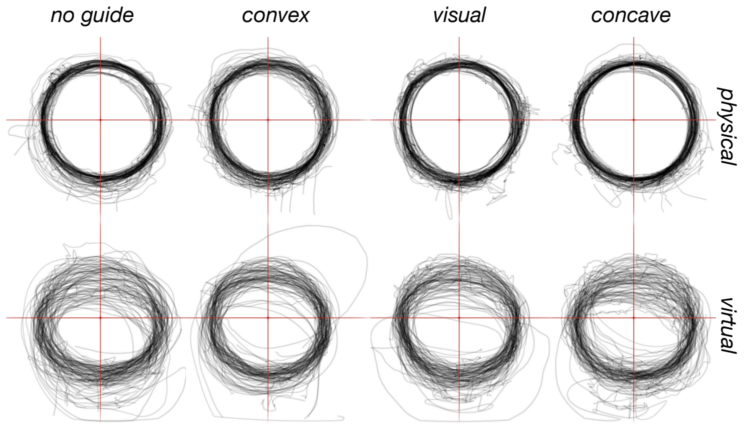
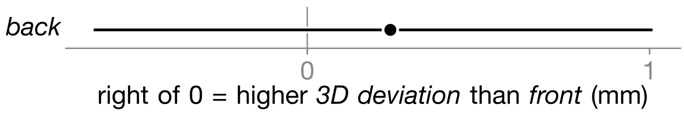


Figure 3.7: Top rendering of the interaction *surface guidance*×*line guidance*. The bottom half of each condition represents the front view. *Physical* strokes show less deviation compared to *virtual* strokes.

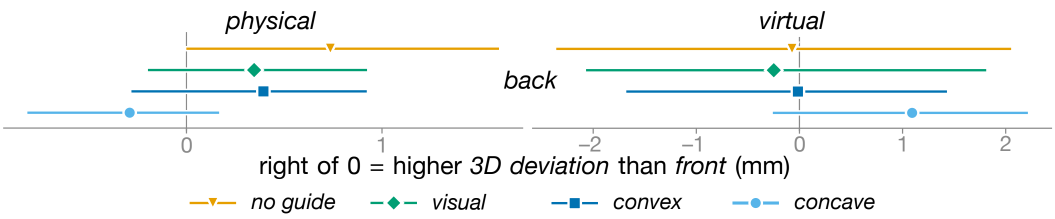
Mean Deviation in 3D

Overall, *side* does not seem to have an effect on the *3D deviation* (*front*: M = 7.0 mm, CI [6.2 mm, 7.9 mm]).



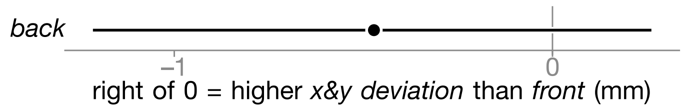
When splitting the data for *surface guidance* and *line guidance*, the *3D deviation* also does not show great differences (*physical*, *front*: M = 4.8 mm, CI [4.3 mm, 5.3 mm]; *virtual*, *front*: M = 9.3 mm, CI [7.4 mm, 11.1 mm]).

Side does not seem to have a consistent effect on accuracy.



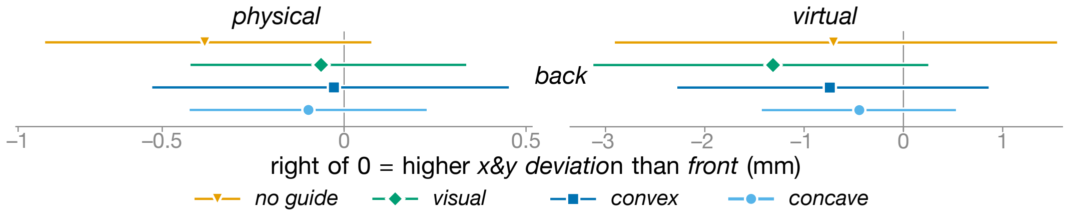
Deviation in x&y Direction

Similarly, there appears also to be no difference for the *x&y deviation* (*front*: M = 5.0 mm, CI [4.4 mm, 5.8 mm]).



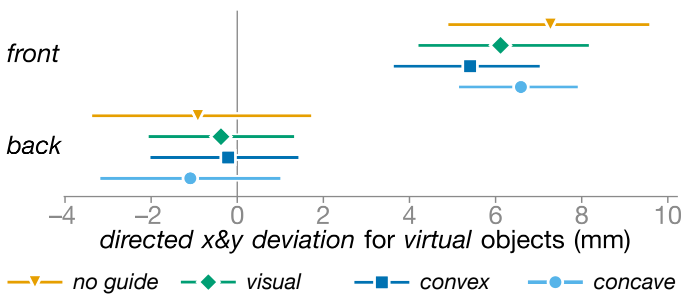
x&y deviation also does not show great differences.

For our individual conditions, there appears only a slight tendency that the *x&y deviation* on *virtual*, *front* is larger compared to the *back* (*physical*, *front*: M = 2.9 mm, CI [2.4 mm, 3.4 mm]; *virtual*, *front*: M = 7.2 mm, CI [5.7 mm, 8.7 mm]).



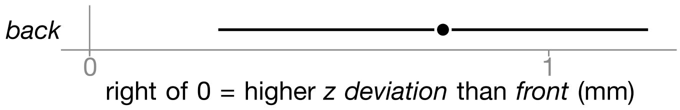
Strokes drawn on the *front* are drawn on the outside for *virtual* objects.

Since the stroke renderings indicate that strokes on the *virtual* objects deviate more into the object on the *back* while deviating out on the *front*, we also calculated the *directed x&y deviation*—points inside the object were given a negative value, points outside a positive value. While the overall means show similar deviation, the directed means indicate that strokes on the *front* are drawn largely outside of the object.



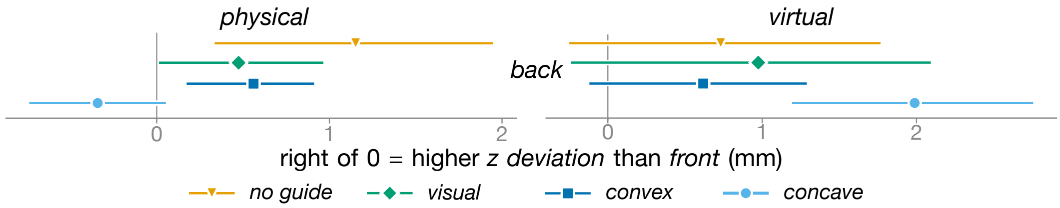
Deviation in z Direction

For the z *deviation*, only a slight trend is visible that overall strokes on the *back* deviate more compared to the *front* (*front*: M = 4.0 mm, CI [3.5 mm, 4.5 mm]).



Looking more closely at the differences shows that particularly *virtual*, *concave* lines seem to be drawn with more z *deviation* on the *back* of objects (*physical*, *front*: M = 3.2 mm, CI [2.9 mm, 3.6 mm]; *virtual*, *front*: M = 4.7 mm, CI [3.8 mm, 5.9 mm]).

Virtual, *concave* shows more z *deviation* on the *back*.



Qualitative Observations

Another intriguing observation is that the *virtual square* shapes showed a slight counter-clockwise rotation (cf. Figure 3.8). Their sides are also traced more accurately than their front and back. Both observations appear in all *virtual* but no *physical* conditions.

Virtual, *square* shapes show a slight rotation.

3.4 Discussion

Our results indicate that both *surface guidance* and *line guidance* affect the performance of drawing on objects. In particular, *physical* objects improve drawing accuracy in all metrics measured. Strokes on *physical* objects deviate less from

Guidance types, esp. *physical*, improve drawing accuracy.

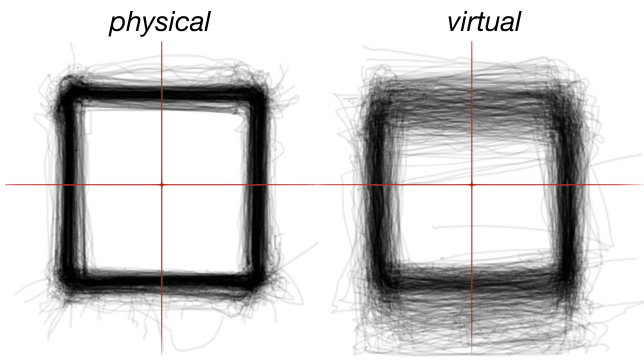


Figure 3.8: Top rendering of the accumulated strokes for the *square* separated by *surface guidance*. The bottom half of each condition represents the front view. *Virtual* strokes appear slightly rotated counter-clockwise and more spread out at the front and back than the sides.

the target, both overall and in each direction. This shows that the hard constraint of a surface supports the user more than its soft "lift-off" constraint.

Line guidance also improves accuracy in most cases.

Evaluating the effect of *line guidance* shows that having *no guide* to follow reduces stroke accuracy. This highlights that any guide at all helps the user to continuously correct deviations from a target line. Especially for *physical* objects, it is an interesting finding that a *visual* guidance performs similar to the guidance given by a physical edge. This means that sketching tools should not only pay attention to the geometry of an object but also its texture.

Especially on *physical* objects, *line guidance* helps to keep the intended height.

The interaction of *surface guidance* and *line guidance* is showing interesting results. On *physical* objects, any guide greatly improves the precision, especially to keep the intended height, while on *virtual* objects, the deviation in height was similar between all *line guidance* conditions. However, that there are no large differences between the different guidance types means that the increased hard constraint by the *physical*, *concave* condition did not have as much impact as we expected. There is only a slight trend that deviation on *physical*, *concave* objects is smaller.

Another interesting finding is that participants performed their stroke on average below the target line on *physical* objects while drawing higher on *virtual* objects. The only exception is the *no guide* condition. As we adjusted the target height to the initial inking height for the *no guide* conditions, this leads to the assumption that participants varied around their initial height but frequently misjudged this height in the *virtual* guide conditions. One possible explanation for this is the issue of detecting depth in virtual environments (cf. Chapter 2.3). As participants were seated at the table and were looking down at the objects on the table, a misjudgment in depth could lead to a higher performed stroke than intended.

Users seem to misjudge the target height on *virtual* objects.

The only effect that the *shape* of an object had on the drawing performance is that it took participants longer to draw a *square* than a *circle*. This may be due to the non-continuous corners of the stroke, which require more attention. The abrupt change in direction likely forced participants to slow down. This is similar to the findings from Cao and Zhai [2007], and Pastel [2006].

Participants took longer to draw around the cuboid.

While our results show differences in the mm range, these already matter for certain modeling tasks, and previous studies indicate that such differences are likely to increase with the size of the target shape [Arora et al., 2017].

The differences will likely increase with larger objects.

After our main evaluation, we also compared the performance on the *front* and *back* of the object, since the stroke visualizations indicated differences. Interestingly, there were no great effects regarding *surface guidance* or *line guidance* for overall or *x&y* deviation. Looking at the top-view renderings suggests that participants deviated more into the object on the back of the object, while drawing outside the object in front. However, looking at the means showed that while many strokes were performed inside the object on the *back*, the directed deviation is close to zero suggesting that the deviation outside of the shape is similar. Since the overall deviation on the *front* is similar to the directed deviation, this means that most strokes were drawn outside of the object. As strokes inside the object were occluded, this means that participants were able to detect whether their stroke was inside the object but had problems determining how far they were away from the surface. Further studies are

The only effect for *side* is that users drew more outside of *virtual* objects on the *front*.

necessary to fully explore the effect of guides under different visibility conditions, and how visualizing both path and guide on the back of objects affects tracing performance.

Depth perception issues could explain the rotation of strokes for *virtual*, *square* objects.

The qualitative observations about a higher accuracy on the sides of a *square* in the *virtual* condition might be explained by the issues humans have with detecting depth in virtual environments (cf. Chapter 2.3). The sides present a clearer edge in $x\&y$ direction, so that participants could judge the correct position more easily. The front and back do not present such a clear border. This can also explain the 'rotation' of squares in the *virtual* condition. As participants follow their stroke on one side, they had to judge whether they were still 'on' the side and continue drawing. The motion that followed was the 'blind' stroke along the back of the object. As participants show the tendency to move into the object, this means that, coupled with overshoots when detecting the edge on the left side, they draw a slightly rotated square. This assumption only explains the offset for counter-clockwise strokes (75.6 % of strokes performed). An intriguing question for future studies is whether the same phenomenon occurs when drawing in clockwise direction. Participants' handedness is also a likely factor in this. Further analysis of the influences of drawing direction and handedness on sketching offsets on *virtual* models thus appear to be an interesting research direction.

Further studies could investigate mental models when using a large spherical tip.

The pen used in our study had a 10 mm diameter spherical tip similar to the pen in the study by Jackson and Keefe [2016], and the stroke drawn originated from the center of this sphere. As explained in the Experimental Design, this meant that it was not possible to draw a stroke visibly on the physical surface. While we applied corrections to the collected data in order to alleviate that effect, further analysis is needed to see whether the performance changes when using a finer tip. Finally, we did not look at the orientation of the pen in this study. Especially with a large, spherical tip, the mental model of where on that sphere the user considers the "drawing point" to be for different surface–pen angles is an intriguing direction for further research.

3.4.1 95 % Neighborhood

Based on the results of our study and the identification of relevant differences, we computed thresholds for various conditions that cover 95 % of the recorded points.

As *surface guidance* had the most influence on drawing performance, we computed the 95 % neighborhood of both *physical* and *virtual* conditions. In the *virtual* conditions, the threshold is twice as large as in *physical* conditions (*virtual*: 22.87 mm; *physical*: 11.15 mm). For *virtual* objects, the difference between *front* and *back* (*front*: 20.34 mm; *back*: 25.05 mm) is greater than for *physical* objects, for which it is practically negligible (*front*: 11.35 mm; *back*: 10.93 mm). The difference for virtual objects is interesting because our evaluation does not show consistent differences in deviation on *front* and *back*, suggesting that participants deviated more constantly on the *back* while having spikes of larger deviation on the *front*.

The area in which 95 % of recorded points lie, is over twice as large for *virtual* objects.

Since *line guidance* had an effect particularly for *physical* objects in the way that any guide at all improved the accuracy, we computed thresholds for those conditions. On *physical* objects with *no guide*, 95 % of the recorded points fall within 14.13 mm of the target stroke while any guide (*visual*, *convex*, *concave*) improves the size of this area to 9.78 mm around the target line.

Line guidance reduces the threshold for *physical* objects.

These thresholds could be used by sketching systems to decide whether a user is still intending to follow a particular line on an object (that the system could then 'snap' the line to) or whether she is intentionally deviating to move to sketching in mid-air.

Thresholds can be used for snapping algorithms.

3.5 Summary & Future Work

We classified the types of guides that existing objects offer for drawing on them. All objects provide *surface guidance* when drawing on them. On *physical* objects, users cannot push into the surface (hard constraint) and should not lift-

We classified guidance types that can guide a stroke.

off the surface (soft constraint). Surface structures like *convex* or *concave* edges guide the user further in drawing along a particular line or curve on the object (*line guidance*). Such lines can also be merely *visual* (e.g., printed). Since AR also allows placing virtual objects into the real world, all guidance types can also exist on *virtual* objects.

We measured the effect of the guidance types. We found that *physical* objects improve stroke accuracy the most.

In our lab study, we quantified the effects these different guidance types have on both accuracy and time needed to complete a stroke when drawing around an object. For this, we synchronized a Vicon motion tracking system with a Microsoft HoloLens, and measured how far participants deviated from the optimal stroke for each guidance type. We found that the deviation was the lowest for the *physical* conditions. In an additional analysis, we found that participants deviated more outside of the *virtual* object in the x&y dimension when the target stroke was on the *front* of the object while being more evenly distributed inside and outside on the *back* of the object.

Future studies could study other drawing tasks or the effect of other object properties.

Our study covers only a small area in the design space of drawing interactions with physical and digital objects in Augmented Reality. We focused on a single-handed task of drawing around a static object. Holding the object to trace in the other hand creates interesting questions regarding the bimanual tracing interactions. These questions include qualitative directions, such as different strategies of performing the stroke, as well as performance related questions, such as the effect on accuracy when performing a bimanual stroke. Further studies should look at the effect of different object orientations, sizes, surface structures, and materials, as well as different pen styles, on drawing performance. AR can also show things that are impossible to see in reality. For example, we occluded the stroke when drawing behind an object, since pilot tests found other approaches to be too confusing but the stroke could still be made visible, e.g., as a dotted line, with potential benefits to the user. Investigating these options and how they can be used to improve modeling tasks in Augmented Reality are promising directions to take in this field.

The system used in this study required expensive equipment and specific calibration to be able to visualize lines drawn in mid-air. Also, using *outside-in* tracking of the mid-air pen reduces the mobility of the whole system. To be able to increase the mobility and make *situated modeling* available for a wider audience, we explored other AR options. Schmalstieg and Höllerer [2016] write that "AR for consumers must be a strict software-only solution delivered to devices the users already own" [p. 411]. This makes Handheld AR an interesting direction to take, since it is supported on most smartphones that people already own. In the next chapter, we describe our ARPen system which makes *situated modeling* possible on consumer smartphones requiring only a 3D-printed pen.

We explored other AR options to make drawing in mid-air more available.

Chapter 4

The ARPen System

Most AR modeling projects use wearable systems and *outside-in* tracking systems reducing the mobility of the system.

We have shown, through the study described in the previous chapter, that sketching on physical objects greatly improves the sketching accuracy in Augmented Reality. This is an important finding for *situated modeling* since it highlights a key benefit of using AR for modeling tasks. The setup we used in our study consisted of a head-mounted Augmented Reality device and an *outside-in* tracking system. Most AR modeling projects use similar wearable systems and tracking setups [e.g., Arora et al., 2018; Peng et al., 2018; Yue et al., 2017]. They offer a hands-free visualization that shows the digital objects overlaid over the real world. However, the reliance on an *outside-in* tracking system reduces the mobility of such setups, and the expensive equipment required limits the availability of *situated modeling* for a wider user base.

Publications: The work presented in this chapter has been done in collaboration with Oliver Nowak, Felix Wehnert, Jan Benscheid, René Schäfer, Simon Voelker, and Jan Borchers. It is in part published as a full paper at ACM CHI '19 [Wacker et al., 2019] and in the master's thesis of Oliver Nowak [Nowak, 2019]. The author of this dissertation is the main author of the paper and developed the research and artifact ideas as well as the motivation. The grasp study was planned together with Oliver Nowak, who also carried out the analysis of results. Most sections in this chapter are taken from the full paper publication. The final section of this chapter contains results from the master's theses of Jan Benscheid [Benscheid, 2019] and René Schäfer [Schäfer, 2020], who worked on their theses under the guidance of the author of this dissertation.

Handheld AR is widely available in consumer smartphones.

Smartphones, on the other hand, are already widely available and have recently gained development support for AR. They can track their position in relation to surfaces in order to place 3D content into their live camera feed. While this requires holding the phone like a camera, moving the phone enables precise viewport control, and AR apps can use the phone touchscreen for interaction. Projecting a 3D scene onto the 2D screen, however, reduces depth information, making it difficult to specify a point at a specific depth in 3D by interacting with the 2D projection [Kruijff et al., 2010; Mossel et al., 2013a; Polvi et al., 2016]. Handheld AR modeling systems have addressed this problem, e.g., by first specifying a plane and then projecting the touch events into it [Xin et al., 2008]. We propose to specify the position of a point in 3D bimanually, using a pen tracked by the smartphone camera. The pen can be held in a *precision grip*, just as a normal pen that is used to draw on paper.

We propose to specify the 3D position with a pen tracked by the phone.

Using the phone's camera for tracking increases mobility.

We chose to track the pen using the smartphone's camera similar to the DodecaPen system [Wu et al., 2017], because it is easy for the user to understand, directly supported in modern AR frameworks, and does not require additional tracking equipment. Other methods of tracking an external input device include, for example, motion capture cameras [Lakatos et al., 2014; Yee et al., 2009], stereo cameras [Milosevic et al., 2016], or by sensing the magnetic field [Yoon et al., 2016]. In Handheld AR, the determined position of an external device has, for example, been used to slice through 3D data [Issartel et al., 2014], control a game by projecting gestures into the 2D gameboard [Hürst et al., 2015], recreate the geometry of a traced object [Milosevic et al., 2016], or for character customization [Seidinger and Grubert, 2016].

The ARPen enables mid-air modeling in Handheld AR with a mid-air pen.

We developed the ARPen system to prototype and evaluate the combination of Handheld AR with a mid-air pen. It uses a recent iPhone and a 3D-printed pen with wireless buttons near the tip and visual markers at its end. The ARPen app tracks the position of the phone in the real-world environment using Apple's `ARKit`, and the position of the pen tip via the pen's visual markers. This allows drawing and interacting with virtual objects that are anchored in real space. Buttons on the pen serve to start and stop drawing, and to invoke other editing commands.

Only requiring smartphone & pen enables this system to be used in many situations, e.g., to design the inset for the can holder in a car to support smaller can sizes, or to combine multiple moving elements such as hinges to create a box.

The screen of the phone adds another input opportunity.

The touchscreen shows the AR scene, and the user can move the phone to change the viewport, e.g., to slice through objects to look behind them. At the same time, the app can use the touchscreen to display interactive controls and options such as a library of model parts that the user can place in the scene and combine with other objects.

The ARPen system is available open-source.

The ARPen system is available in the iOS App Store[1] to make it available for exploration for a multitude of users. Furthermore, we have made the code available open-source on Github[2] as a toolbox for researchers to prototype and evaluate new interaction techniques in the area of Handheld AR with a mid-air pen without having to set up the whole system from scratch.

To better understand this new interaction technique, we first analyzed how people would hold a smartphone while drawing with the pen in the other hand, and which parts of the screen they are capable of reaching with the same hand (Chapter 4.3). We then present two improvements for the ARPen system that have been explored in student theses: one that extends the system with a modeling kernel for the creation of complex geometries, and one that evaluates different placements of the markers on the pen (Chapter 4.4).

In summary, we make the following contributions in this chapter:

- the ARPen interaction technique that allows for bimanual *situated modeling* in Augmented Reality using a pen and consumer smartphone,
- the ARPen toolbox for easier access to Handheld AR with a mid-air pen, both for users through the App store as well as developers through the open-source release,

[1] hci.rwth-aachen.de/arpen-ios (accessed 28.05.2021)
[2] github.com/i10/ARPen (accessed 28.05.2021)

- user evaluation of the grip and reachable touchscreen areas in this technique,
- thesis summaries describing the inclusion of a modeling kernel with the implementation of basic geometry creation techniques, and the design and evaluation of different marker placements.

4.1 Interaction Technique

The combination of mid-air pen and phone results in a *bimanual asymmetric* interaction.

Specifying a position in mid-air is important for 3D modeling applications. A precise 3D input device—similar to the mouse in desktop settings—is needed to perform controlled strokes and manipulations in mid-air. The hand-held device provides another area of interaction: the touchscreen to switch modes or adjust settings, and the device itself by setting the viewport. The combination of both pen and phone results in a *bimanual asymmetric* technique.

We want to study the combination of pen+phone.

Each of those areas of interaction have tasks suited to their strengths: precise 3D input capabilities for the pen, familiar haptic interaction on the touchscreen. However, the combination pen+phone has the potential to improve and simplify 3D model generation in Augmented Reality. We want to use the ARPen system to prototype and study these combinations. For example, the hand holding the device could select an object while the pen is used to manipulate the object's rotation or size. Or the touchscreen could be used to select a model which is placed using the pen. Similar to J. H. Lee et al. [2018], manipulating the viewport while holding the object with the pen could scale an object. Sequencing actions such as a translation followed by rotation could also be faster by switching the action on the screen while the pen adjusts the value.

4.2 Implementation

Tracking the position of the, in most cases pen shaped [Jackson and Keefe, 2016; Arora et al., 2018], input device in rela-

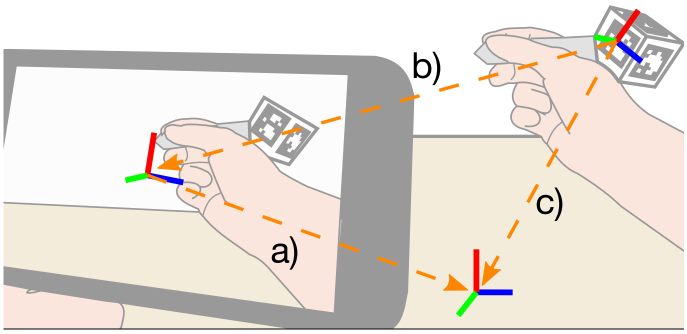


Figure 4.1: a) `ARKit` is used to determine the camera's pose relative to the surfaces. b) `arUco` tracks the marker relative to the camera. c) The combination allows calculating the position of the marker relative to the surfaces.

tion to the world is essential for mid-air modeling. We implemented a tracking requiring only a smartphone. With marker tracking, such as with `arUco`[3] [Garrido-Jurado et al., 2014], we are able to track the position of a marker in camera coordinates. Wu et al. [2017] do this with a fixed camera. However, `ARKit` for iOS allows tracking the device's *view transform* relative to surfaces in the scene. Combining the two techniques, we can track the 3D location of a marker in the world as shown in Figure 4.1 (cf. Chapter 2.2.2 for more information on the tracking techniques). The ARPen has six `arUco` markers on its end. This ensures that at least one marker is visible for the camera even if the pen is pointing away from the camera. Knowing the physical setup of pen and markers, we can determine the pen tip from the marker location and stabilize this location by averaging, if multiple markers are visible. Furthermore, the pen transmits the states of three buttons via BLE.

We combine `ARKit` with marker tracking to determine the position of the pen in the world.

We wrote an iOS app to calculate the mid-air position of the pen. The initial implementation has been done by Wehnert

[3] docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html (accessed 28.05.2021)

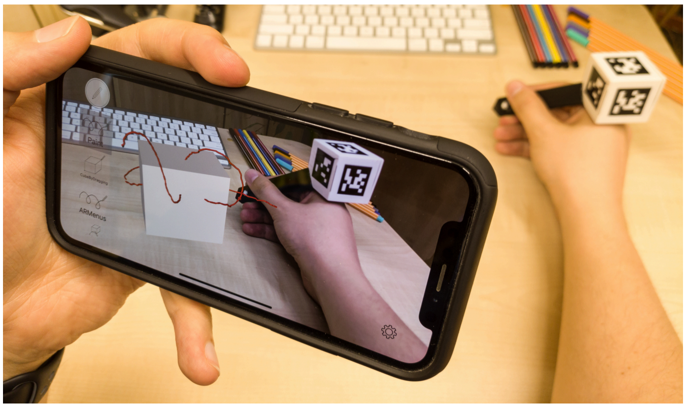


Figure 4.2: With the ARPen system, it is possible to draw lines in mid-air or specify the extend of a cube by using a pen tracked with a consumer smartphone.

The iOS app allows drawing in mid-air and adding new interactions.

[2018] for his bachelor's thesis under the guidance of the author of this thesis. Using `SceneKit`[4], we can render a sphere at the tip of the pen. The implementation makes it simple to define new interactions based on the 3D position of the pen tip and the pen's button states. For example, holding a button and moving the pen could draw a path mid-air or define the diagonal of a cube (cf. Figure 4.2). To enable others to analyze their own mid-air modeling techniques, we provide an open-source implementation of the ARPen system on Github[5].

4.3 Phone Grip & Interaction Area

We studied how people hold the device and which areas are reachable on the screen.

When using the ARPen system, the phone is operated with the non-dominant hand while the dominant hand is used for the pen interaction, making the interaction with the ARPen a *bimanual asymmetric task* [LaViola et al., 2017, pp. 433–435]. In such tasks, the frame of reference—in our case the viewport—is controlled by the non-dominant hand, and the fine-grained interaction is performed by the dominant hand. We conducted a study on how people hold the

[4] developer.apple.com/documentation/scenekit (accessed 28.05.2021)
[5] github.com/i10/ARPen (accessed 28.05.2021)

phone and which areas they can reach on the screen while drawing in mid-air using the ARPen.

4.3.1 Study Setup

We included different *size*s and *orientation*s.

Participants used the ARPen on smartphones of different *size*s: 4.7″ iPhone 6s (*small*) and 5.5″ iPhone 7 Plus (*big*). The *orientation* in which the participants should hold the phone varied between a *portrait* and two landscape orientations differing on the position of the device's camera. Holding the phone in the left hand, the hand is either grabbing the top (*camera left*) or the bottom of the screen (*camera right*). Both the order of *orientation* and *size* were counterbalanced.

We recorded touches on the screen to determine interaction areas.

No menu elements were shown on the screen to not influence the grip of the participant by the placement of buttons. We recorded the position of all touches on the screen to find out the available interaction area. This recording could be activated by the moderator to avoid accidental recordings while adjusting the grip.

4.3.2 Study Procedure

Users had to find a comfortable grip and touch available areas on the screen.

In the beginning of each condition, the phone was placed in front of the participant and she was told in which orientation it should be held. The participant was asked to draw freely in AR with the pen and find a suitable grip of the smartphone for the current orientation. After the participant said that they found a comfortable grip, we took a photo of how the phone was held and started the 'touch recording' mode. The participant had to trace and fill out the area she could reach with a finger of the hand holding the phone. No visual feedback was shown to avoid influencing the participant. We then stopped the recording, cleared the drawing area, and placed the phone back on the table. The sequence was repeated for all different *orientation*s. Then, the conductor changed the phone to the other *size* and the task was repeated. Afterward, the participant was asked about the preferred *size* and *orientation*.

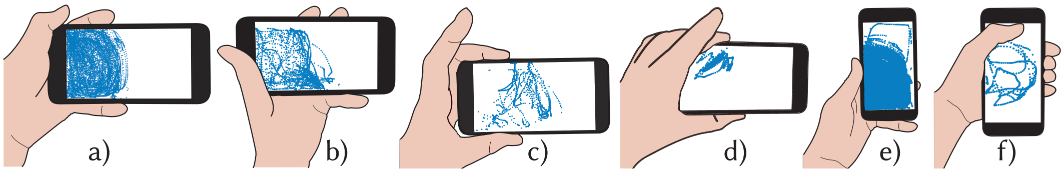


Figure 4.3: Grasps found in the study (with touch recordings). a) *pinkie*, b) *thumb tray*, c) *frame*, d) *front*, e) *low portrait*, f) *high portrait*. *Pinkie* and *low portrait* were used most often for their respective orientation.

4.3.3 Evaluation

We split grasps into *valid* and *invalid* grasps and analyzed touch points for *valid* grasps.

Our initial categorization of grasps was into *valid* grasps that allowed touching the screen and *invalid* grasps in which no interaction with the screen was possible—either because all fingers were involved in holding the device and could not be lifted or touching the screen caused the phone to fall down. For *valid* grasps, we analyzed the recorded touch points. Touch points of participants who held the phone in the right hand were mirrored to allow for a combined evaluation.

4.3.4 Results

18 participants (4 female, 2 n/a) took part in the study (M = 25.7 years, SD = 3.0 y). One left-handed participant and one right-handed participant held the phone in the right hand.

Size

The preferences regarding *size* are very balanced.

Nine participants preferred the *big* phone because of the larger screen and because they felt that a wider camera image simplified keeping the pen's markers in view. Seven participants preferred the smaller phone and mentioned a more comfortable grasp and less weight. The remaining participants had no preference.

Orientation

Size did not affect the subjective ratings regarding the *orientation*. *Portrait* and *camera right* both were most preferred similar times (*portrait*: 9, *camera right*: 8). However, *portrait* was placed in last place four times because the limited horizontal viewport would cause losing the pen's markers. *Camera left* was rated lowest by 14 participants, because most common grasps would occlude the camera.

Portrait and *camera right* are the most preferred *orientation*s.

Grasps

We categorized the *valid* landscape grasps in *pinkie*-, *thumb tray*-, *frame*-, and *front*-grasp (cf. Figure 4.3, a–d):

A *pinkie*-grasp was used most often in landscape orientation.

- *Pinkie*: The phone rests on the pinkie finger, with the index finger holding the top.
- *Thumb tray*: Similar to the *pinkie* but the phone also lies on the thumb tray.
- *Frame*: Thumb and middle finger form a frame by holding the phone from the side.
- *Front*: The phone is held with the thumb, index and middle finger laterally from the front.

Pinkie was used most for *camera right* especially (cf. Table 4.1). It was used less in *camera left* as it often occluded the phone's camera and participants adjusted the grasp towards *thumb tray*. We classified eleven grasps as *invalid*—mostly because the participant used a *frame*-grasp but with the index holding the top, leaving no finger to touch the screen. For *portrait*, participants used two *valid* grasps, both using the thumb for interaction: for *low portrait*, the phone is held on the bottom, and for *high portrait*, the phone is held around the middle (cf. Figure 4.3, e&f).

		small	big
camera right	*pinkie*	50 %	61.1 %
	thumb tray	16.7 %	11.1 %
	frame	11.1 %	11.1 %
	front	0 %	5.6 %
	invalid	22.2 %	11.1 %
camera left	*pinkie*	27.8 %	27.8 %
	thumb tray	33.3 %	27.8 %
	frame	11.1 %	16.7 %
	front	11.1 %	16.7 %
	invalid	16.7 %	11.1 %
portrait	*low portrait*	100.00 %	88.9 %
	high portrait	0.00 %	5.6 %
	invalid	0.00 %	5.6 %

Table 4.1: Grasp frequencies for different orientations and sizes. *Pinkie* and *low portrait* are used most often.

Reachable Areas

Touch points for *pinkie* and *thumb tray* are to the bottom left of the screen; for *frame* and *front*, they are in the top center

The recorded touch points show that for landscape the reachable area of *pinkie* and *thumb tray* is located to the bottom left, while *frame* and *front* are located in the top center (cf. Figure 4.3, a–d). We grouped those grasps together and defined general touch areas by calculating average boundaries in x and y direction. For *pinkie* and *thumb tray*, the touch area has a width of 59.2 mm (*big*: 50.9 mm), a height of 52.2 mm (*big*: 56.0 mm) and is 0.8 mm away from the left edge and 2 mm from the bottom of the screen (*big*: 2.9 mm left, 5.1 mm bottom). *Frame* and *front* were closer to the screen center—20.4 mm from the left and 0.0 mm from the top of the screen—with smaller average width and height of 51.7 mm and 43.9 mm (*big*: 19.5 mm left, 2.89 mm top, 53 mm width, 41.1 mm height). Touch points for *low portrait* are 0.9 mm from the left and 2 mm from the bottom of the screen (*big*: 1.5 mm left, 1.9 mm bottom) with a width of 54.0 mm (*big*: 54.4 mm) and height of 66.7 mm (*big*: 68.3 mm). Lastly, for *high portrait*, which was used once on the *big* phone, the touches are 1.6 mm left and 20.55 mm from the bottom of the screen and the area has an average width and height of 58.9 mm and 78.3 mm.

4.3.5 Summary

We used a *big* iPhone and the *pinkie*-grasp in the *camera right* orientation for the further studies. The *camera right* orientation was never the least preferred orientation as *portrait* had issues keeping the pen markers in view. The *big* phone gave participants the impression of seeing more of the scene also reducing issues of keeping the pen in view. Controlling this grasp avoids finding interaction techniques that works well only in a screen orientation that is otherwise not working well for designing.

We use the *pinkie*-grip in future studies.

4.4 Extending and Improving the ARPen System

Our initial implementation of the ARPen system enables users to draw lines in mid-air and the study in the previous section provides a first impression of the interaction with this asymmetric bimanual system. However, in order to create more complex geometries needed for *situated modeling* tasks, the capabilities of the system need to be expanded. Also, the initial design of the ARPen features the marker cube on the back of the pen. This might not be the best position as it splits the focus of the user between the tip of the pen and the back of the pen. The author of this thesis worked together with students to address these questions and the findings have been published in the master's theses of Jan Benscheid and René Schäfer. Benscheid [2019] extended the modeling capabilities of the system by including the `OpenCascade` modeling kernel into the ARPen. This allows the ARPen system to perform CAD tasks and enables the development and evaluation of techniques to create more complex models. Schäfer [2020] designed and evaluated different placements of the marker cube to determine which is preferred by users.

We extended the ARPen system with a modeling kernel and evaluated different placements of the marker cube.

The author of this thesis worked together with the students on the design decisions and provided advice and feedback for study design and evaluation. Therefore, we summarize the results here.

4.4.1 Modeling Kernel Integration

`SceneKit` does not offer advanced modeling operations.

So far, the ARPen system is using the `SceneKit` framework to create objects and show them in AR. However, `SceneKit` does not offer advanced modeling operations such as calculating intersections or cuts of different objects, as well as the creation of more complex geometries. To address this, we explored ways to extend the modeling capabilities. This work is also published in the master's thesis of Benscheid [2019].

`OpenCascade` was integrated into the ARPen in a layered architecture.

Aside from enabling the creation of models from sketches, a modeling kernel for the ARPen system also needs to be able to be run on iOS hardware without additional tethering and be available free and open-source as well. After comparing different modeling kernels, we decided to focus on `OpenCascade`[6] for the ARPen system as it fulfills our requirements and is a long-existing and supported framework. Benscheid included the `OpenCascade` framework in a layered architecture. He implemented bridges for the communication between the `SceneKit` layer used to render the objects and the `OpenCascade` layer responsible for the geometry generation. This structure allows combining the implementation of basic interaction on the `SceneKit` layer with the calculation of advanced geometry operations without great changes to the current plugin architecture. In his thesis, Benscheid [2019] provides a more detailed explanation on the different layers and their connection.

We designed techniques for sweeps, revolutions, lofts, and boolean operations.

Besides drawing lines, Benscheid identified sweeps, revolutions, lofts, and boolean operations as the most frequently used modeling features based on prior work, e.g., by Kang et al. [2015]. For these operations, we designed different interactions on how to control them. Note that the goal is not to create a fully functional modeling system but to demonstrate the capabilities of the system and explore interaction techniques that allow the creation of volumetric models.

[6] www.opencascade.com/ (accessed 28.05.2021)

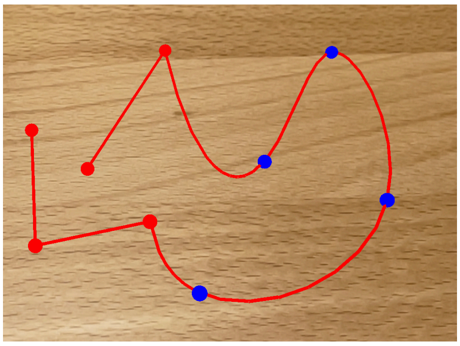


Figure 4.4: A path generated by fitting a line through control points. A red control point creates a *sharp* corner, while a curved line is created through blue control points.

Drawing Lines

A path can be created by setting control points.

The naive version to create lines in the initial ARPen implementation generates many points connected by cylinders. To reduce the number of points and reduce the influence of jitter, we decided to enable line creation through control points. The final line is fitted through all control points and the user can set for each control point whether the line is taking a sharp corner at the control point or whether a curved line is created from the previous control point through this control point to the next. For this, the user can toggle between a *sharp* or *curved* control point when creating the path (cf. Figure 4.4). When coming close to the starting point of the path, the visualization snaps to a closed path that the user can confirm by placing a control point. When holding down either control point creation button, new control points are placed at regular intervals to simulate freehand drawing of a path. Future advancements could also analyze freehand strokes to filter out the jitter and calculate an approximate path.

Sweeping

Sweeping: creating a volume by moving a surface along a path.

Sweeping describes the creation of volumes by moving a surface along a path. A special case of a sweep is *extrusion* in which the profile is swept along a straight line perpen-

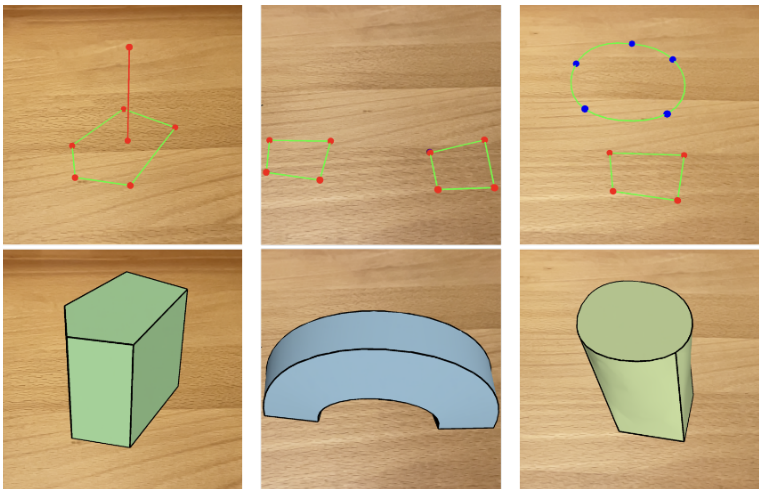


Figure 4.5: Sweeping techniques. *Left*: *Profile&path*, *Middle*: *Second profile*, *Right*: *Loft*.

dicular to the profile, e.g., to create a cube. In our sweeping techniques, we focussed on sweeps of flat profiles along an arbitrary path.

Profile&path: users specify a closed and an open path.

Sweeping by path (*profile&path*): The first sweeping technique for the ARPen system is controlled by specifying two elements: a closed path and open path. Once both paths are drawn, the system replaces them with the object created by sweeping the closed shape along the open path (cf. Figure 4.5, left).

Second profile: users specify the start and end profile.

Sweeping by two profiles (*second profile*): Another option to create a sweep is by specifying not only the start profile but also where the final object should end. After specifying both profiles, the system calculates the connection between them and creates the object. If a straight connection is not possible, the system creates a curved connection that is minimizing the bending (cf. Figure 4.5, middle). However, as it is not possible to completely duplicate the closed shape, we decided to use the initial shape as the definition of the shape and use the second profile to specify the position and orientation of the final form. The minimum requirement to specify this information is the creation of a plane. Therefore, only three points have to be given to specify the endpoint of the sweep.

This interaction can be improved by enabling the user to move a copy of the initial profile to the desired end location of the sweep. This requires interaction techniques to move and rotate objects, which we take a closer look at in Chapter 6. Another shortcoming of the current implementation is that it is not possible to specify the sweeping path after the object has been created. An updated version could show the path and allow editing of the path.

The option to copy and place the starting profile could improve this technique.

Lofting (*loft*): Another potential adaption to the problem of the end position of the sweep is to use lofting. Lofting is similar to sweeping in that a profile is swept along a profile. However, unlike general sweeping, the profile can change shape along the sweeping path to result in a different profile at the endpoint. We designed a technique similar to the *second profile* sweep in which the user can specify two closed shapes and the system will create a volume morphing from one shape to the other (cf. Figure 4.5, right). Similar to the *second profile* sweep, the current implementation does not allow the user to refine the calculated lofting path after the object has been created.

Loft: the created object morphs between the start and the end profile.

Revolution

Revolution describes the operation in which a shape is rotated around an axis to create a volume. This can also be seen as a sweep of a profile with a circular sweeping path ending at the same location. For our designs, we allowed both open and closed profiles for the revolution. If the user specified an open profile, we added a lid and bottom to create a fully closed object.

Revolution: creating a volume by rotating a shape around an axis.

Revolution around an axis (*profile&axis*): Similar to the profile and path for the sweeping interaction, this interaction is using the basic information necessary to create a revolution. The user has to specify two paths, one to define

Profile&axis: the user creates a path and a rotation axis.

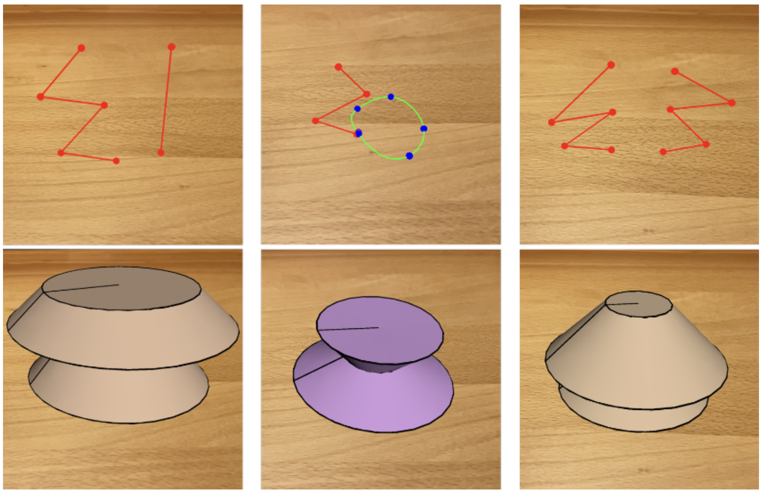


Figure 4.6: Revolution techniques. *Left*: *Profile&axis*, *Middle*: *Profile&circle*, *Right*: *Second profile*.

the profile that should be rotated and one straight path to specify the axis around which the profile should be rotated. Once both paths are specified, the system calculates the final object (cf. Figure 4.6, left).

Profile&circle: a path is rotated around an axis specified by a circle.

Revolution along a circle (*profile&circle*): As mentioned, revolution can also be interpreted as sweeping along a circular path. This interaction follows that by taking a profile and a circular path as input for the generation of the object. As it is hard to specify a perfectly circular path and to differentiate this technique from the sweeping technique, the control points of the circular path are used to compute the best fitting circle through them. The center of this circle is used to define the rotation axis for the revolution (cf. Figure 4.6, middle).

Second profile: an additional path defines the extend of the object.

Revolution with a second profile (*second profile*): Similar to the sweeping with two profiles, a second profile can also be used to specify the revolution. For this, the user can draw a second profile to describe the outline of the final object. The system then defines the rotation axis by calculating the middle between the two profiles (cf. Figure 4.6, right). Again, as it is not possible to exactly replicate the first profile, only the required information is taken from the second profile. This means that only the start and endpoint of the second profile are matched with the start and endpoint of the first profile to determine the rotation axis.

Boolean Operations

In *constructive solid geometry*, boolean operations are used to combine existing objects to create more advanced objects [Foley et al., 1996]. Two main operations are *union* (or *merge*) and *difference* (or *cut*). *Union* means that the objects are grouped and fused together. For example, four rectangular blocks could be combined with a wider flat block to create the model of a simple table (cf. Figure 4.7, left). On the other hand, *difference* means cutting the geometry of one object out of the geometry of another object. An example could be to cut a sphere out of a cube to create a bowl (cf. Figure 4.7, right). We designed and implemented two different ways how these boolean operations can be realized with the ARPen system. Note that the use of boolean operations requires techniques to select and move objects to arrange them. We present and evaluate techniques to achieve this in Chapter 6.

Boolean operations are used to combine objects into more complex ones.

Boolean operation through *order*: For the initial way, the user has to select both objects after they have been arranged. By pressing one of two buttons on the pen (or on the touchscreen), the user can perform either the merge or cut operation. For the cut operation, the object that has been selected first will be cut out from the second.

Cut and merge buttons are used to combine objects.

Boolean operation by *object state*: Another way to control the boolean operation is by specifying the state of the object before the combination. The state can either be *solid* or *hole*. When two objects have been selected and the "combine" operation is called, the objects are combined based on their current states. If two *solid* objects are combined, the resulting object is the union of the objects. If a *solid* object and

The state of the object can be set before combining.

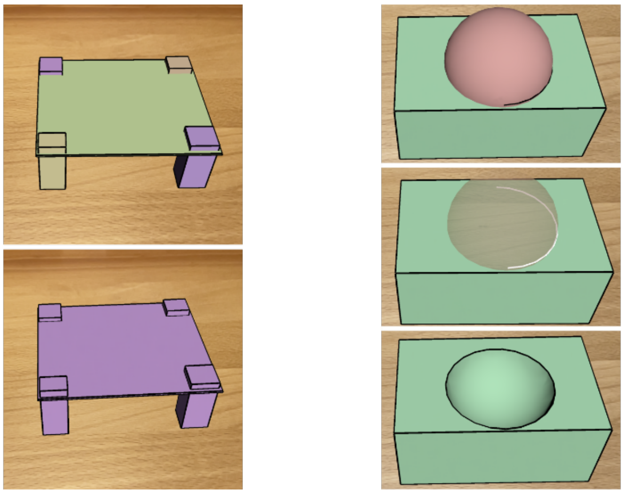


Figure 4.7: Boolean Operations. *Left*: *Union* of five elements to form a table, *Right*: *Difference* between a sphere and a cuboid by changing the sphere to *hole* before combining.

a *hole* object are combined, the *hole* object is cut away from the *solid* object (cf. Figure 4.7, right).

Evaluation of Modeling Techniques

Benscheid evaluated the sweeping and revolution techniques.

Benscheid [2019] implemented the designed techniques and evaluated the sweeping and revolution techniques with twelve participants to gather both initial feedback on the techniques, as well as general insights into the requirements for modeling tools. For his study, he asked participants to recreate models of objects shown to the participant. The models differed in the complexity of both the profile and the path (for sweeps) as well as profile and radius (for revolutions). As the focus of the study was on the operations generating the volumetric model, Benscheid provided the initial profile for each object. He gathered quantitative measurements including *task completion time*, a *model quality* calculated by deviation from the ideal model, as well as subjective ratings in form of *ease-of-use* Likert scales and a *ranking* of the used techniques. He also asked questions to gather qualitative comments regarding the different techniques and the overall system.

Benscheid found in his study that participants seem to prefer interaction methods that require the least input. For the sweeping techniques, users preferred the *profile&path* technique if only a simple extrusion was needed but the *second profile* technique if the sweeping path got more complicated. In this case, users seemed to have trouble specifying the correct orientation for the final steps of the path. On the other hand, participants mentioned for *second profile*, that adapting the automatically determined path would be desirable. Using the *second profile* technique, users predominantly tried to copy the original profile if it was a simple form, while approximating the more complex profile mostly with a triangle. For the revolution techniques, interestingly users preferred the *profile&axis* technique and disliked the *second profile* technique even though the quantitative measurements indicate a lesser model quality for the *profile&axis* technique. While most participants used a simple line to specify the position of the second profile, it might be harder to anticipate the final result compared to the rotation around an axis. On the other hand, an error in specifying the rotation axis directly results in a more skewed object compared to an error in specifying the second profile as the rotation axis is calculated together with the original profile.

Users prefer methods that require little input but want control over automated steps.

To analyze the qualitative comments throughout the study, Benscheid grouped the comments into clusters. The overall clusters he found were comments about *depth perception*, *visual guides*, *virtual constraints*, *sweep specification*, and general comments regarding *hardware*.

Benscheid clustered the qualitative comments.

***Depth Perception*:** The biggest concern of the participants were issues in detecting the correct depth (11/12). This showed, for example, in problems in finding the start point of a path to close it. These issues are also apparent in the created models in that objects are “leaning” even though an orthogonal extrusion was planned. Also linked to this, users mentioned that they preferred drawing on a physical surface (7/12).

Depth perception was a big issue.

***Visual Guides*:** Showing more information for the current interaction created another cluster of comments. The

Users requested live feedback.

participants mentioned problems estimating the sizes of shapes and objects (8/12) and suggested live previews of the current operation (7/12) or displaying a grid and current measurements (5/12) as potential solutions.

Snapping could improve the techniques.

***Virtual Constraints*:** Making the previous point more explicit, participants also mentioned that it would be helpful if the input would *snap* to relevant points (10/12). This included comments of snapping to existing points to close a path (4/12) or to specific angles to allow e.g., for orthogonal extrusions (4/12). Two more comments also mention snapping to existing objects to be able to place control points on them more easily.

Users were unsure about the result of a sweeping operation.

***Sweep Specification*:** The comments regarding the cluster *sweep specification* address the two sweeping techniques. A big issue of participants was the creation of curved paths (9/12) and in particular that they were uncertain of the orientation of the final profile at the end of the path (3/12). On the other hand, participants mentioned feeling uncertain about the sweep that would be created with the *second profile* (4/12). As one reason, participants said that it was not clear how the center of the second profile is determined (3/12) and others said that they would like to be able to add intermediate profiles to control the sweeping path more (2/12).

Users mentioned issues of keeping the markers on the pen in view.

***Hardware*:** The biggest concern for the *hardware* concept was the trouble of keeping the markers on the back of the pen within the viewport to allow for drawing (9/12). We address this concern in more detail in the next section and in Schäfer [2020]. The other comments for this concept are not directly linked to the modeling techniques and more about the overall system usage. This includes comments about having to switch between mid-air and touchscreen input with the pen hand (4/12), suggestions to use AR glasses (3/12), comments about the heat of the phone (3/12), trying to focus the camera by tapping the screen (2/12), and feeling fatigue in the arms (2/12).

Figure 4.8: A cube with markers at the back of the pen allows the pen to be tracked in almost every hand position. Only pointing the pen at the camera causes markers to be occluded by the hand.

In summary, none of the techniques performed consistently worse than comparable techniques. The preference appears to be more dependent on the use case, indicating that users should have multiple interaction methods available so that they can choose the best for the current situation. A trend from the user evaluation is that participants seemed to prefer the technique that required the least input but, on the other hand, reduced the amount of "automatic" calculation for which the result was not clear. As expected, depth perception is a large issue for the interaction with a system such as the ARPen and many comments from the participants can be seen as a result of these issues. Offering immediate feedback during the interaction and using visualizations to increase the understanding of the spatial relationship between objects are important factors for this interaction. We look at different visualization techniques to improve moving to a specific point in Chapter 8.

Users seem to prefer techniques that require little input but reduce "automatic" calculations for which the result is not clear.

4.4.2 Evaluating Marker Cube Positions

Keeping the markers of the pen in view is one of the issues found in the modeling study by Benscheid. This issue occurs because users have to split their attention between their point of interest, which is the tip of the pen, and the back of the pen to check that the markers are still in the visible frustum of the camera. Having the markers on the back of the pen also has benefits, as at least one marker is visible for nearly every hand posture (cf. Figure 4.8). Another issue with the initial pen design is that the electron-

Users have to split their attention between the tip of the pen and the back of the pen.

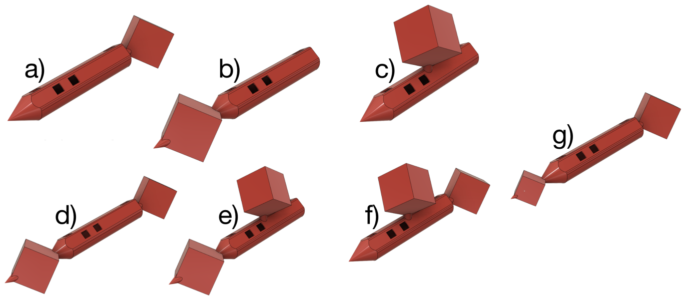


Figure 4.9: Placements of the marker cube. a) *Back*, b) *Front*, c) *Top*, d) *BackFront*, e) *TopFront*, f) *BackTop*, g) *BackFrontSmall* (adapted from [Schäfer, 2020])

ics such as the bluetooth chip and battery are placed inside the marker cube. While this enables easy access to them, it also increases the weight at the back of the pen, potentially causing an unfamiliar weight distribution.

Schäfer redesigned the electronics of the ARPen.

Schäfer [2020] addresses both issues in his thesis, which he worked on under the guidance of the author of this thesis. First, Schäfer redesigned the electronics and wiring to fit inside the stem of the pen, distributing the weight more equally and freeing up the marker cube to be moved around more freely. We refer to his thesis for further details on this redesign [Schäfer, 2020].

We created pens with the marker cube at different positions.

Second, we created seven pen designs in which the marker cube is located at different positions (cf. Figure 4.9). Aside from the current position at the *back* of the pen, we also placed it on the *front* and on the *top* of the pen. For the pen with the marker cube at the *front*, we included a small tip extruding from the cube to clearly mark the position of the pen tip. Placing the marker cube at the *front* reduces the distance between the pen tip and the markers, however, it also increases the possibilities for occlusion in different hand positions. Therefore, we also created combined versions with two marker cubes by adding either a *back* cube or a *top* cube to the pen with the *front* cube (*backFront* & *topFront*). Completing these combinations, is a pen with both *top* and *back* cube (*backTop*).

The size of the marker cubes' sides is 3 cm for all cubes as we found that this size produces good tracking quality in our setup, even if the arm is outstretched and further away from the camera. The quality of the tracking is also dependent on the quality and resolution of the camera. In our setup, smaller cubes were showing more issues with the tracking performance. However, when the pen is held closer to the camera, a smaller cube would also be sufficiently large in the camera view for tracking. Since the issues with the marker cube at the *back* occur if the back of the pen leaves the camera view, we expect that the pen is held closer to the camera in these cases. Therefore, we also designed a hybrid pen with the standard cube at the *back* for the tracking in larger distances and a smaller, 1.8 cm cube at the *front* for situations in which the *back* cube is not visible (*backFrontSmall*). Reducing the size of the cube at the *front* reduces the distance between the pen tip and the hand holding the pen, potentially feeling more like a normal pen than with the larger *front* cube.

A pen with a small cube at the front and a larger cube at the back could combine the benefits of both designs.

To gather information from users about the different pen designs, we designed a task in which participants had to trace along an object with different surface features—specified with our guidance types presented in Chapter 3.2—and perform a general movement in mid-air (cf. Figure 4.10). Schäfer [2020] conducted the user study with 28 participants. Following the completion of the task with a pen, the participants had to fill out a questionnaire measuring the *subjective perception* of the design. After having used all pens, participants were asked to create a *ranking* of all designs.

Users had to trace over an object using the pen designs.

The results of the *ranking* of techniques (cf. Figure 4.11) align well with the other subjective information gathered from different questionnaires. In all, the *top* pen achieved the highest rating and preference followed by the other *top* designs *backTop* and *topFront*. Other designs with the cube in the *front* achieved the lowest ranking and ratings.

The *top* pen achieved the highest ratings before the *back* and *front* designs.

The biggest complaints of participants regarding the *front* cube were that they occluded the markers with their own hands (15/28) and had troubles drawing around the object (12/28). Also the cube would occasionally block the view

Users often occluded the *front* cube with their hands.

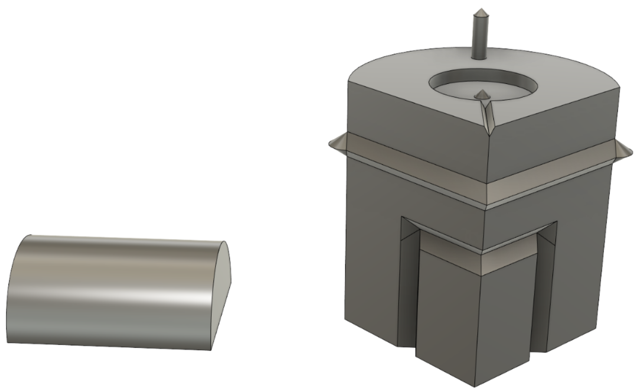


Figure 4.10: Participants had to trace the large object following *concave* and *convex* surface guides before having to connect the objects with a mid-air line. The task included drawing operations in many directions and under different surface conditions (Image: [Schäfer, 2020])

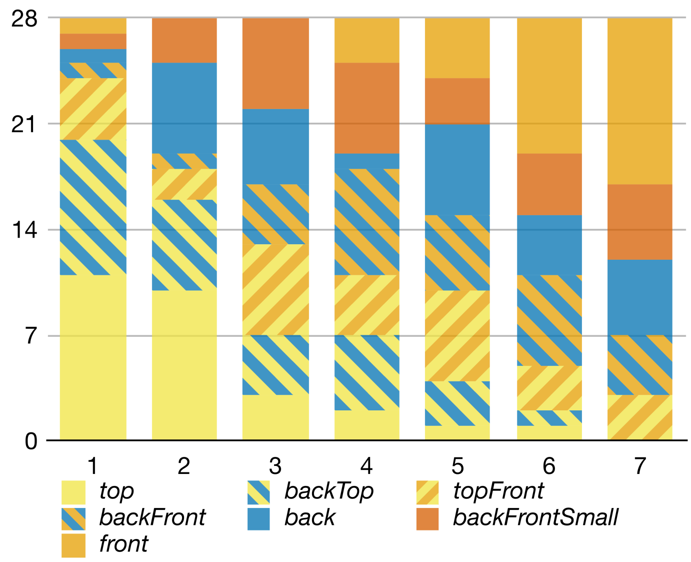


Figure 4.11: Subjective ranking of pen designs. The horizontal axis represents placements in the ranking (1=best) and the vertical axis indicates how often a design has been placed at this position in the ranking. The ranking shows that designs with a *top* cube have been preferred by users while designs with a *front* cube were mostly placed on the lower side of the ranking.

to the tip of the pen (5/28), or was in the way (4/28). While participants liked the idea of the smaller cube (11/28), the cube would still be in the way (5/28).

The *top* pen is the preferred design.

In summary, the cube on *top* of the pen seems to be the best mix between reducing the distance between pen tip and markers while not obstructing either pen tip and marker and still achieving tracking in most hand orientations.[7]

4.5 Summary

We introduced the ARPen system, enabling sketching in Handheld AR with a mid-air pen.

In this chapter, we have introduced the ARPen system, which enables sketching in Handheld AR with a mid-air pen, requiring only a current smartphone and a 3D-printed pen. Such an asymmetric bimanual interaction offers many interesting directions for research, for example, how the input capabilities of the touchscreen could be combined with the mid-air position of the pen. The ARPen system is available in the iOS App Store but is also available open-source to allow others to implement and evaluate interaction techniques for mid-air pen input in Handheld AR. In an initial study, we evaluated how users hold the device while sketching in mid-air and what parts of the screen are still in reach. We found that a *pinkie*-grip in landscape orientation was preferred by our participants, leaving the side of the screen available for interaction with the thumb. Over the course of two master's theses, the ARPen system was extended with a modeling kernel that allows for the creation of complex geometries, and the 3D-printed pen was improved by moving the marker cube necessary for tracking closer to the pen tip. With these capabilities, it is already possible to design objects directly in Augmented Reality.

A *pinkie*-grip is preferred by users.

Many basic interactions are not defined for such a system.

During our early experiences using the ARPen, it became apparent that many basic interactions are not clear for a system combining Handheld AR on a smartphone with a mid-air pen. What are suitable interaction techniques to arrange

[7] The thesis of Schäfer and the evaluation of marker placements was carried out after most of the other studies included in this thesis. Therefore, every study except for the scaling study (Chapter 6.4.2) is using the *back* design.

objects in the scene? How can we best use the different components of such a system? To build a solid understanding of the interaction space of Handheld AR with a mid-air pen, we decided to focus on exploring the essential interactions in 3D systems. In the next chapter, we describe a classification of mid-air interactions by LaViola et al. [2017] before evaluating different essential interactions in the following chapters.

Chapter 5

Mid-Air Interactions

In the previous chapter, we have introduced the ARPen as the implementation of a Handheld AR system with a mid-air pen. We found that many basic techniques are not defined for such an asymmetric bimanual interaction system, and decided on designing and evaluating these basic interaction techniques. In this chapter, we cite a categorization of essential interactions in 3D user interfaces, and we use this categorization in the following chapters of this thesis to present our interaction techniques and evaluations.

We present a classification of essential interactions in 3D user interfaces.

Mid-air interaction covers a wide variety of areas and topics. However, there are several categories that occur in most 3D interfaces. LaViola et al. [2017] name three central categories: *selection & manipulation*, *system control*, and *navigation*. In this chapter, we briefly describe these categories based on information from their textbook [LaViola et al., 2017, part IV, pp. 251–418]. Even though LaViola et al. mostly focus on immersive AR and VR systems and less on Handheld AR, their classification is still applicable to our scenario and helps in structuring the approach to address a wide variety of interactions for a novel interaction category.

The three categories are *selection & manipulation*, *system control*, and *navigation*.

5.1 Selection and Manipulation

Selection means picking a specific object in the scene.

Selection & manipulation covers the main interaction with virtual objects in the application and findings regarding these interactions can be applied to other interaction techniques. *Selection* is the first basic action that LaViola et al. mention. This means picking a specific virtual object in the scene, for example, to apply further changes to this object. To achieve this action, the user has to (1) target the intended item and (2) confirm the selection of the target. In desktop operating systems, this is handled by moving the mouse cursor over the target and pressing the mouse button. In Chapter 6.2, we present an evaluation of different *selection* techniques for Handheld AR with a mid-air pen.

Manipulation is the direct interaction with an object.

Whenever a user directly interacts with an object and applies changes to it, probably a *manipulation* action is applied. LaViola et al. narrow the definition of *manipulation* to be about *spatial rigid object manipulation* that does not change the shape of objects [LaViola et al., 2017, p. 257]. According to LaViola et al., the appropriateness of a manipulation technique also depends on the use-case of the application—a suitable technique for one application scenario does not have to be suitable for every other scenario. Throughout this thesis, we design and evaluate our interaction techniques with our use-case of *situated modeling* in mind. In this use-case, the manipulation techniques can, for example, be used to arrange virtual objects in order to combine them using boolean operations [Foley et al., 1996]. However, the user studies in the following chapters feature also very generic tasks, so that our findings can be applied to different application scenarios. Furthermore, we focus on *canonical manipulation tasks* mentioned by LaViola et al. that include basic tasks that are used to form more complex interactions, instead of manipulation tasks that are specifically designed for one application.

Translation means moving an object to a new location.

The first manipulation action is *translation*. This means picking up an object and moving it to a new location. Users have to both be able to specify which object to translate and to specify the location the object should be moved to. We discuss different designs of *translation* techniques and their evaluation in Chapter 6.3. Closely linked to this is the action of *rotation*. Instead of the position, this action addresses the orientation of the object, such as tipping a cube on its side. Chapter 6.4.1 contains initial information on *rotation* techniques in Handheld AR with a mid-air pen. *Scaling* is the one basic manipulation technique that does not occur naturally in the real world as it is in most cases not possible to change the size of a physical object. In virtual environments, however, the size of an object such as a box can be increased or decreased arbitrarily. We cover *scaling* techniques in Chapter 6.4.2.

Rotation means changing the orientation of an object.

Scaling means adjusting the size of an object.

LaViola et al. also mention that manipulation tasks in virtual environments can be separated into *isomorphic* and *non-isomorphic* techniques [LaViola et al., 2017, pp. 262–263]. *Isomorphic* techniques aim to closely replicate the real world action. For the example of moving an object, an *isomorphic* translation technique would be to grab the object with the hand and drop it at the new location. *Non-isomorphic* techniques use the more “magical” properties and possibilities of virtual systems. Instead of moving to the object, grabbing it with the hand, moving to the target position, and dropping it there, the user could also point at the object she wants to move to make it fly to her hand. LaViola et al. say that studies have shown that *non-isomorphic* techniques often achieve better results than *isomorphic* techniques as they can simplify actions that are not possible to change in the physical world. Especially for our use-case of drawing and modeling in mid-air, *isomorphic* control of the drawn stroke appears to be the best option as it allows using the physical properties of the real world directly. However, for the interaction with created objects and performing the basic manipulation tasks, *non-isomorphic* techniques should be considered. In the studies carried out with the ARPen System—presented in the following chapters—we include both *isomorphic* and *non-isomorphic* techniques.

Isomorphic techniques replicate the real world action; *non-isomorphic* techniques use the “magical” properties of virtual systems.

5.2 System Control

The second category of interaction in 3D interfaces is *system control*. For LaViola et al. [2017], *system control* is defined by

System control tasks issue commands to the system, e.g., by using graphical menus.

tasks that issue commands to the system or provide symbolic input. These commands are issued to the system to either (a) trigger a particular function, (b) change the interaction mode, or (c) change the system state. While *manipulation* techniques specify both "what" should be done and "how" it should be done, the system is carrying out the task in *system control* tasks, meaning that only the "what" is triggered by the user. "System control is critical because it is the glue that lets the user control the interaction flow between the other key tasks in an application" [LaViola et al., 2017, p. 380]. However, according to LaViola et al., *system control* tasks have not been as heavily researched. One of the main tasks for *system control* is the interaction with graphical menus. We present different menu designs for the different interaction methods for Handheld AR with a mid-air pen in Chapter 7.

In Handheld AR, text entry can be achieved by typing on the touchscreen.

Symbolic input, such as text entry, is another big category for the interaction with a system. However, since Handheld AR often uses a smartphone or tablet as the screen, the question of symbolic input is more clear as it is possible to use the touchscreen of the device to type-in text. On head-worn AR and VR systems, this is a bigger issue and we refer to LaViola et al. [2017] for more information on how this can be achieved in those systems.

5.3 Navigation

The third category of interaction is the *navigation* inside the system. LaViola et al. [2017] name two subtasks of the *navigation* task: *travel* and *wayfinding*.

Travel is the motor component of *navigation*.

Travel describes the motor component of moving to a new position or in a certain direction. In the real world this translates to simple walking or using assisting elements such as a steering wheel to specify where to move. According to LaViola et al. [2017], travel techniques are central for the interaction, for example, as they enable the users to look around. Furthermore it is very important to make the travel interaction easy to use so that the user can be immersed in the system and does not need to think about how to move around. As many VR systems are limited to a smaller physical space, this requires special techniques, such as redirected walking or teleportation, to cover larger distances. Especially in VR, the actual travel movement is often carried out by the computer in the end. Since the interaction area in a Handheld AR system with a mid-air pen for drawing and modeling is generally smaller, the motor component is a lesser issue. The viewport can be directly adjusted to a new position—called the *camera-in-hand* technique [LaViola et al., 2017, pp. 349–350]—and the tip of the physical pen represents the virtual cursor specifying an absolute position in the scene.

In Handheld AR, the viewport can be controlled directly and the pen is moving around the scene.

Wayfinding represents the cognitive component of the *navigation* task. This includes both the perception of the current position in the virtual environment, as well as understanding the required steps to perform to get to a new location. Therefore, this component includes tasks such as spatial understanding, planning, and building a map of the environment. The system provides *wayfinding aids* to assist in perceiving the space, being able to formulate where to go, and other tasks in the user's mind. In our context of Handheld AR with a mid-air pen, the orientation component is very important. Due to the perceptual issues in Handheld AR (cf. Chapter 2.3), interpreting the position of the pen and the virtual objects is not easy. However, being able to move the pen tip to a specific position defined by a virtual object is necessary to be able to start drawing a path from the correct position. *Non-isomorphic* drawing techniques, which project the starting point of a drawing to the object hit by a ray-cast, could address this issue but would lose the benefit of the assistance of using the physical guidance in the environment (cf. Chapter 3). In Chapter 8, we design and evaluate visualization techniques to improve the perception of the pen tip's position in relation to other objects in the scene.

Wayfinding represents the cognitive component of *navigation*.

The classification by LaViola et al. [2017], splitting interactions in 3D interfaces into *selection & manipulation*, *system control*, and *travel*, can assist in covering relevant interaction with a new system. In the next Chapters in this thesis, we address basic interactions for each of the categories to achieve first insights in the interaction with a Handheld

In the next chapters, we study basic interactions for each of the categories.

AR system with a mid-air pen, as well as show the potential study possibilities with the ARPen system. For *manipulation*, we design and evaluate techniques to select, translate, rotate, and scale virtual objects (Chapter 6). In Chapter 7, we present different techniques for context menus to interact with the system. For the *navigation* task, we focus on *wayfinding* by investigating visualization techniques that assist in perceiving the position of the mid-air pen in relation to virtual objects in the scene (Chapter 8).

Chapter 6

Evaluating Selection and Manipulation Techniques

To interact with virtual objects in AR or VR requires *selection & manipulation* operations, in order to specify an object to change its properties, or to move it to a new location. For our usage scenario of *situated modeling* in Handheld AR with a mid-air pen, these actions could be used to pick up a virtual model and position it relative to another model so that one can be cut out from the other.

For *situated modeling*, objects need to be arranged in the scene.

To evaluate the interaction with virtual content in the scene, we performed two studies comparing different techniques for two of the central tasks when interacting with AR content: selecting and translating virtual objects. We then look

We designed and evaluated *selection & manipulation* techniques.

Publications: The work presented in this chapter has been done in collaboration with Oliver Nowak, Donna Klamma, Farhadiba Mohammed, Simon Voelker, and Jan Borchers. It is in part published as a full paper at ACM CHI '19 [Wacker et al., 2019]. The author of this dissertation is the main author of the paper. He also developed the research ideas, designed and implemented the selection and translation techniques as well as planned the user studies. He reanalyzed the data for this thesis. Most sections in this chapter are taken from the full paper publication. The final section of this chapter contains results from the bachelor's theses of Donna Klamma [Klamma, 2019] and Farhadiba Mohammed [Mohammed, 2020], who worked under the guidance of the author of this dissertation.

at designs and studies of rotation and scaling techniques that have been worked on in bachelor's theses under guidance from the author of this thesis.

In this chapter, we make the following contributions:

- design and evaluation of selection techniques for Handheld AR with a mid-air pen,
- a user study evaluating different translation techniques to move objects to a new position,
- design of rotation and scaling techniques, and summaries of user studies by thesis students comparing these techniques.

First, we look at related work on manipulation techniques for Handheld AR systems.

6.1 Related Work

In Handheld AR, many techniques use the touchscreen for interaction.

In Handheld AR, many interaction techniques involve the touchscreen to manipulate the virtual content. For object selection, intuitive methods are to directly touch the projection of the object, or to have a central crosshair on the screen [Vincent et al., 2013]. However, touching the screen for a selection often moves the device and thus the viewport into the scene. This can cause selections errors. To address this, several approaches 'freeze' the camera feed during touch interaction [e.g., G. A. Lee et al., 2009; Bai et al., 2012; Vincent et al., 2013], which improves accuracy but takes longer. Special techniques improve selection in dense environments [Mossel et al., 2013b].

A key limitation of touchscreen interaction is the missing dimension.

A key limitation of Handheld AR is that the touchscreen does not directly support manipulating the six degrees of freedom of a virtual object in space. 3DTouch by Mossel et al. [2013a] addresses this by interpreting swipes on the screen in the context of the position of the smartphone in the world: E.g., looking at an object from the front (in the

$-z$ direction) and dragging the projection moves it in the $x\&y$ plane, while looking from the top (in $-y$ direction) moves it in the $x\&z$ plane. For rotation and scaling, the interaction on the touchscreen is similarly interpreted depending on the position of the smartphone. Another approach is to first let the user tap to cast a ray into the scene, fix the object to be placed on that ray, then drag on the screen to move the object along the ray [Polvi et al., 2016].

Movement of the device can be used for interaction.

Device movement can also be used to transform the virtual object [e.g., Henrysson et al., 2005; Mossel et al., 2013a; Hürst and Wezel, 2013; Polvi et al., 2016; Marzo et al., 2014] or move a cursor around a 3D environment [Babic et al., 2018]. Selecting and holding an object attaches it to the device. Now any device movement is applied to the object, enabling compound manipulation of position and rotation, for example, with the HOMER-S technique by Mossel et al. [2013a]. Studies show that users prefer this technique over touchscreen interactions, achieving good translation and rotation performance [Mossel et al., 2013a; Marzo et al., 2014]. However, for scaling tasks, interaction on the touchscreen with the 3DTouch technique was preferred to the HOMER-S technique [Mossel et al., 2013a].

Tracking the user's hands enables gesture input.

A more natural approach tracks the user's free hand while holding her phone, using a separate depth camera [Bai et al., 2014; Billinghurst et al., 2014] or the device camera [Bai et al., 2012; Hürst and Wezel, 2013]. Colored finger markers improve the tracking [Hürst and Wezel, 2013], although this project only tracked 2D movements and used device motion for the missing dimension. At the time, although users enjoyed the interaction, gesture tracking was not accurate enough for fair evaluation [Bai et al., 2012]. Goh et al. [2019] provides a more detailed review of object manipulation techniques in Handheld AR.

The ARPen enables direct 3D input.

Unlike these approaches, the ARPen system allows direct 3D input suitable for modeling without multiple steps or viewport adjustments, and the pen buttons support issuing commands without touching the screen.

So far, no project compared different techniques for the manipulation of virtual objects in Handheld AR with a pen.

We present detailed studies evaluating selection and translation techniques before presenting results from student theses that the author of this work provided assistance for.

We designed and evaluated selection and translation techniques.

For the selection and translation techniques, we ran two studies that required participants to interact with objects within a 40 cm×40 cm×40 cm volume on a table surface. This size enabled participants to stay seated during the study while holding phone and pen but they were encouraged to move the phone or stand up if they wanted to.

6.2 Selection Study

For selection, the user has to target an object and confirm the selection.

The selection of objects is required for manipulations of a specific object, such as a change in color or a transformation. In order to select an object, the user must first target the object and then confirm the selection. For a survey of object selection techniques in Virtual Reality, see Argelaguet and Andujar [2013].

We compared five *selection technique*s.

We compared five different techniques of selecting an object in mid-air using the ARPen system (*selection technique*), combining different targeting and confirmation methods. We measured the success rate, selection time, deviation from the target, and the size of the object on the screen during the selection.

6.2.1 Selection Techniques

We sampled fundamental techniques to perform the selection from a large space of possible interactions. We designed Pen Selection Without Highlighting (*without highlight*), Pen Selection With Highlighting (*with highlight*), One-handed Touch Selection (*one-handed*), Two-handed Touch Selection (*two-handed*), and Pen Ray Selection (*pen ray*). See Figure 6.1 for details.

Without highlight: no virtual visualization of the position.

Pen Selection Without Highlighting: In this technique, the position of the pen is not visualized in the scene. Users have to match the pen tip position in the real world to the

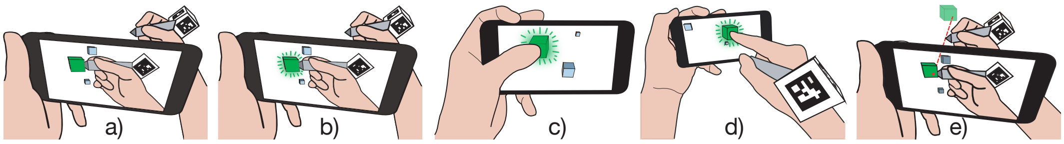


Figure 6.1: Selection techniques. a) *Without highlight*, b) *With highlight*, c) *One-handed*, d) *Two-handed* as the baseline, e) *Pen ray*.

position of the virtual object. Pressing a button on the pen confirms the selection at the current position.

Pen Selection With Highlighting: Here, the tip of the pen is visualized with a red sphere. The sphere disappears if the pen tip is behind or inside an object. If it is inside an object, the object changes into a visually highlighted state. *Without highlight* measures how well the selection worked just from the standard visualization and from using triangulation of a mid-air position with the *camera-in-hand* movements. *With highlight* evaluates how basic depth cues could help improve the depth specification. A button press on the pen confirms the selection.

With highlight: an object changes its color if the pen tip is inside it.

One-handed & Two-handed Touch Selection: Since the virtual objects are displayed on top of the live camera feed on a smartphone, the touchscreen can also be used for the interaction. From the position of a touch on the screen, a ray is cast into the scene and the first virtual object it contacts is selected. For *one-handed*, the user has to select the object using the thumb on the hand holding the phone. In the *two-handed* condition, the user performs the selection with the hand holding the pen.

One-handed & *two-handed*: the user touches the object on the screen.

Pen Ray Selection: Upon pressing a button on the pen, a ray is cast through the tip of the pen and the first target it hits is selected, similar to *occlusion selection* techniques in VR [Argelaguet and Andujar, 2013]. This means that the user has to align the tip of the pen so that it is in front of the intended target. Pressing the button on the pen then selects this target behind the pen tip.

Pen ray: the object behind the pen tip is selected.

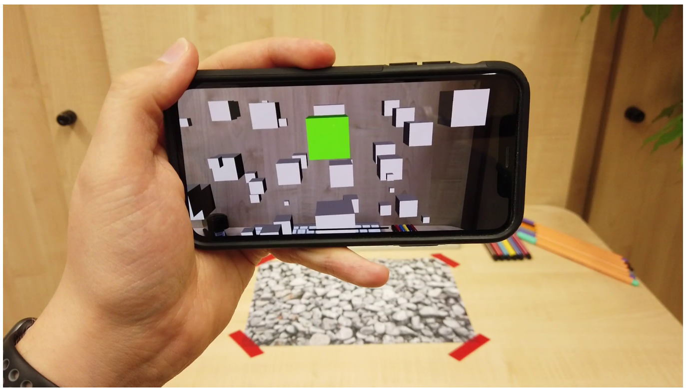


Figure 6.2: In the selection study, virtual cubes were placed in a grid in front of the user. The cube the user had to select was highlighted.

6.2.2 Study Setup

We distributed 64 cubes in the interaction volume.

The interaction volume was separated into 64 cubic areas with an edge length of 10 cm (*cube space*). Into each of these *cube spaces*, we placed a white cube with an edge length of either 1 cm, 2 cm, 3 cm, or 4 cm. For each trial, 16 cubes of either size were assigned randomly to the *cube spaces* and their position within each *cube space* was randomized but needed to be at least 1 cm away from its side. Therefore, the distance between cubes was at least 2 cm (cf. Figure 6.2).

Participants had to select 64 targets with each technique.

Participants had to select 64 targets using each of the five *selection technique*s. At the beginning of a trial, one cube was shown as the target by changing its color to green. After a selection occurred, regardless of whether it was correctly selected or not, a new cube was shown as the target. The order of targets was randomized.

6.2.3 Study Procedure

Participants had to use the *pinkie*-grip.

Participants sat in front of a table with tape markings to improve the world tracking and a visual marker to keep the position of the interaction volume consistent between participants. Each participant was asked to hold the phone in her non-dominant hand using the *pinkie*-grasp. She was given time to familiarize herself with the grip and the ARPen before we introduced the task and techniques.

We demonstrated and explained each technique before letting the participant try for herself. For *two-handed*, the participant had to hold the pen in the hand tapping the screen to stay in the scenario of modeling in mid-air before selecting an object. Once the participant was confident, we started the trial and asked the participant to select the 64 cubes one after the other. If the participant needed a rest during the study or stopped to make a comment, we restarted the last selection. After selecting all 64 cubes, we asked the participant to rate the *ease of selection* and *confidence of selection* when using the technique. For each technique, we noted qualitative observations about the selection strategy. After using all techniques, the participant was asked to rank the five techniques from best to worst. Overall, each participant selected 320 targets (64 cubes$\times$5 *selection technique*s). The order of conditions was counterbalanced between participants using a Latin square.

We asked participants to rate the techniques.

6.2.4 Measurements

Beside recording the success of the current selection, we measured the time from showing the target to the issued selection. For not successful selections, we calculated the deviation to the target. For *without highlight* and *with highlight* this deviation is the length of the vector between the specified 3D point and the target (in cm). To evaluate the offset for each dimension, we stored the direction of this vector in camera coordinates. For *one-handed*, *two-handed*, and *pen ray*, we measured the distance from the selection position on the screen to the convex hull of the target's projection (in mm). Since moving the phone adjusts the size of the target on the screen, we recorded the size of the bounding box of the projection at the time of the selection (in cm^2). We also collected subjective ratings for *ease of selection* and *confidence of selection* for each technique on 7-level Likert-Scales and a ranking of the five techniques.

We measured success, time, deviation, and subjective ratings.

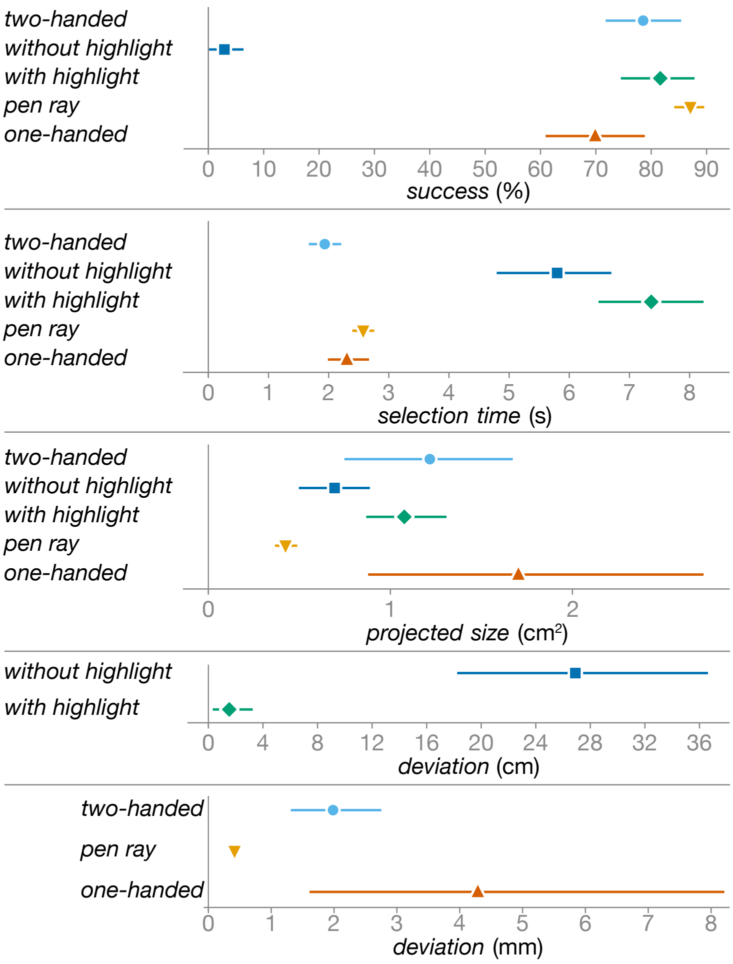


Figure 6.3: Results of the selection study. Means and 95 % CIs without within-subject adjustments.

6.2.5 Results

We recruited 15 participants (1 female, 1 n.a., 21–40 years, M = 28 years, SD = 5.4 years, all right handed). We discarded one participant's *without highlight* data as the selection was intentionally performed differently. Overall, we recorded 4735 selections. For every participant, we counted the successful selections per condition and divided it by the number of trials (*success*) as well as averaged the deviation for missed selections (*deviation*), time (*selection time*), and projected size (*projected size*).

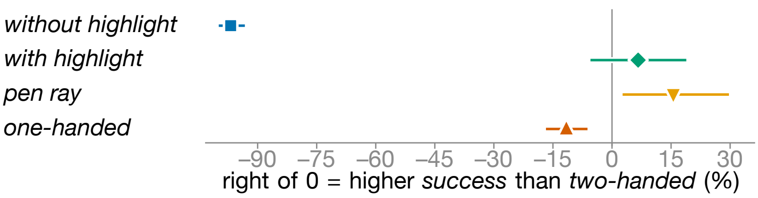
We calculated the confidence intervals with the method explained in Appendix A. The overall confidence intervals without within-subject adjustment are shown in Figure 6.3. The following sections contain the within-subjects mean differences to estimate the performance of using a particular *selection technique* compared to the baseline of *two-handed*. This means that for the upcoming plots, a value to the right of '0' suggests more e.g., successful selections compared to the baseline. For *success*, we did not subtract the percentages of the technique and the baseline but divided the success rate of the technique by the success rate of the baseline before calculating the differences. Therefore, we are able to report the differences not as percentage points but percentages. Also, as differences for Likert Scale ratings are not defined, we did not compute the difference to the baseline but report the means and bootstrapped confidence intervals. Individual results such as the measurements for the overall confidence intervals, are calculated and reported the same way. Compared to the findings reported in the published paper [Wacker et al., 2019], this evaluation shows the same major findings.

We analyze within-subject differences using confidence intervals.

Success

The plot below show the percentage difference regarding successful selections compared to the baseline (*two-handed*, M = 78.5 %, CI [71.8 %, 85.4 %]). It becomes clear that participants had great issues selecting objects with the *without highlight* technique and almost no object was selected correctly. The other techniques are closer together. Only *one-handed* seems to be a bit below the performance from the remaining technique with a trend that *pen ray* performs better compared to *two-handed*.

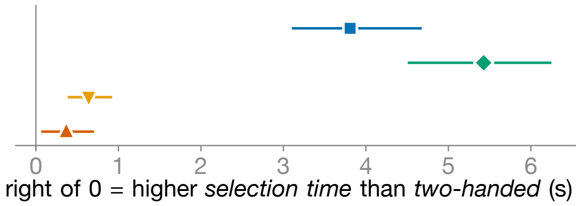
Without highlight has by far the lowest success rate.



Selection Time

With highlight & *without highlight* require the longest selection time.

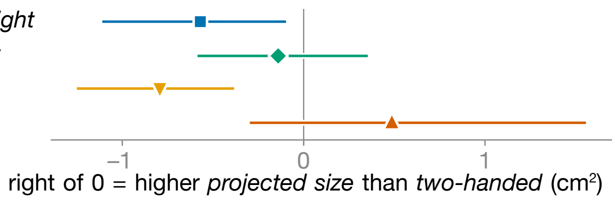
Regarding *selection time*, *two-handed* was the fastest on average with *one-handed* and *pen ray* not a lot slower (*two-handed*, M = 1.9 s, CI [1.7 s, 2.2 s]). Participants clearly took longer with the *without highlight* technique with *with highlight* being the slowest.



Projected Size

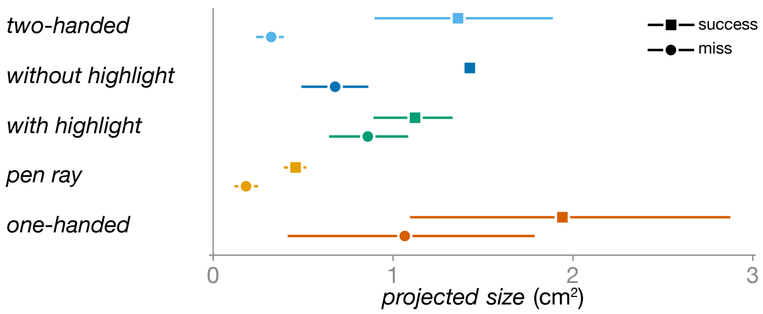
Targets seem to be smaller for *pen ray* & *without highlight*.

The *projected size* at the point of selection shows only small differences. Targets seem to be the smaller for *pen ray* and *without highlight* compared to the baseline (*two-handed*, M = 1.2 cm^2, CI [0.7 cm^2, 1.7 cm^2]). *With highlight* shows similar target sizes with a trend that targets were larger when using the *one-handed* technique.



The *projected size* is larger for successfully selected targets.

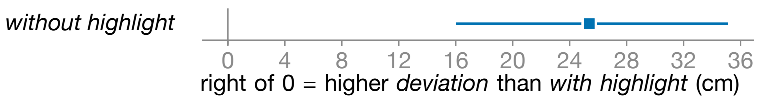
Splitting the results based on *success* shows that *projected size* for successfully selected targets is larger than for misses. This difference seems to be smaller for *with highlight*. For all other conditions, *projected size* is about twice as large or more for successful selections.



Deviation

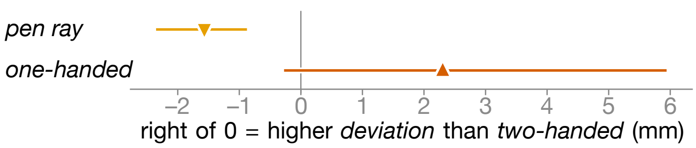
The *deviation* in the mid-air pointing techniques was larger for *without highlight* compared to *with highlight* (*with highlight*, M = 1.5 cm, CI [0.3 cm, 3.3 cm]). The average deviation vector in camera coordinates shows a large offset in *z*-dimension (*without highlight* x: 7.1 cm, y: 2.7 cm, z: 27.4 cm; *with highlight* x: 0.5 cm, y: 0.7 cm, z: 2.4 cm).

Without highlight has the highest deviation.



For the screen selection techniques, we recorded the least *deviation* for *pen ray* followed by *two-handed* and *one-handed* (*two-handed*, M = 2.0 mm, CI [1.3 mm, 2.8 mm]).

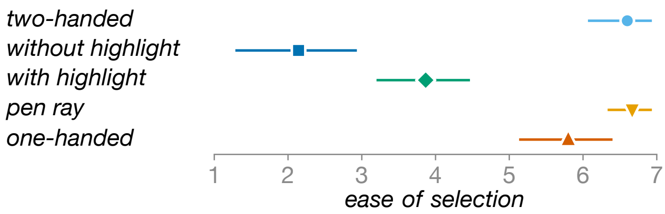
Pen ray has the lowest deviation.



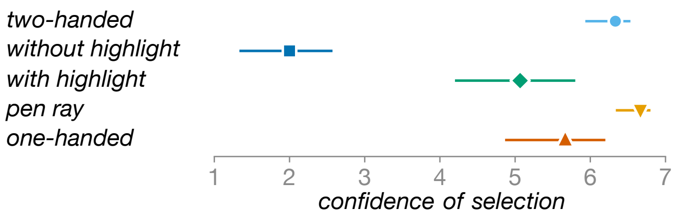
Subjective Ratings

We found that for *ease of selection*, both *pen ray* and *two-handed* achieved high scores followed by *one-handed*. *With highlight* received lower ratings with *without highlight* being rated the lowest.

Pen ray & *two-handed* get high subjective ratings.



The ratings for *confidence of selection* are similar to the ratings for *ease of selection*. The biggest difference is that *with highlight* achieved ratings much closer to the high rated techniques.



The ranking shows that participant liked both *pen ray* and *two-handed* followed by *one-handed*. *Without highlight* is ranked mostly as the least preferred technique trailing *with highlight* (cf. Figure 6.4).

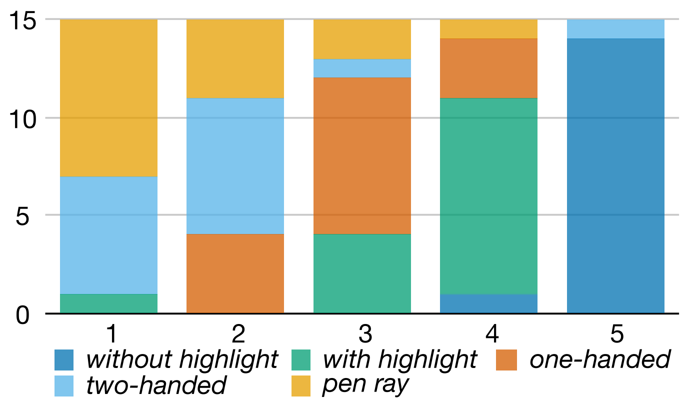


Figure 6.4: Subjective ranking of *selection technique*. The horizontal axis represents placements (1=best) and the vertical axis indicates frequency of placement at this position. *Without highlight* is least preferred. *Pen ray* and *two-handed* are on the first two places.

Qualitative Remarks

A recurring qualitative remark suggests that participants understood the arrangements of the boxes better in the *with highlight* condition compared to other conditions. For *pen ray*, most participants used a strategy in which they did not vary the distance between pen and phone but moved them together. Overall, observations indicate that the device was moved more in the *one-handed* and *two-handed* conditions. As expected, participants mentioned fatigue in both their arms from holding them up for an extended period of time, indicating that this interaction technique should only be used for shorter durations.

Participants state that they understand the scene arrangement better in the *with highlight* condition.

6.2.6 Discussion

Pen ray seems to be the best candidate for selection tasks with the ARPen. This technique has the highest *success* rating combined with a quick *selection time*. The small *projected size* indicates that the device did not have to be moved much to select targets. Together with *two-handed*, *pen ray* is also the preferred selection technique of participants. *With highlight* has a good *success* rate but the *selection time* was the slowest likely because participants had to adjust their position to the correct depth for the selection. However, this might lead to a better understanding of the arrangement of objects in the scene. The touch conditions performed well based on the *selection time* and *two-handed* also shows a good *success* rate. The *projected size* indicates the phone was moved closer to the targets which might become more exhausting over time. *Without highlight* had the least *success* and was the least preferred technique. This shows that depth perception in Handheld AR requires additional feedback and that *non-isomorphic* interaction techniques were preferred to *isomorphic* techniques. For the next study, we did not consider a condition without visual feedback.

Pen ray and *two-handed* achieve the best results.

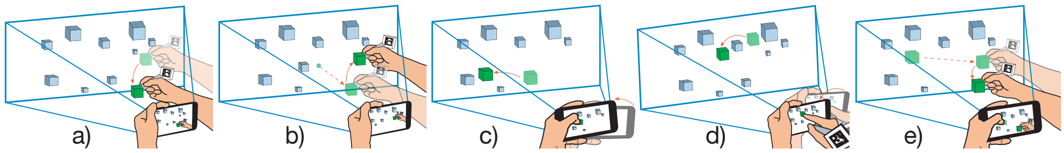


Figure 6.5: Translation techniques. a) *Pen drag&drop*, b) *Pen ray pickup*, c) *One-handed*, d) *Two-handed* as the baseline, e) *Touch&pen*.

6.3 Translation Study

We study the interaction of selecting and moving a virtual object.

Selection is in many cases only the starting point of a manipulation. We studied the whole interaction of selecting an object, moving it through the scene, and dropping it at a different location in this study. In 2D operating systems, this corresponds to a drag & drop operations.

6.3.1 Translation Techniques

We designed 5 *translation techniques*.

We selected five *translation technique*s to drag and drop a virtual target in a 3D environment using Handheld AR with a mid-air pen. Four are based directly on selection techniques from the previous study: *pen drag&drop*, *pen ray pickup*, *one-handed*, and *two-handed*. *Touch&pen* combines touchscreen and pen (cf. Figure 6.5).

Pen drag&drop: continuation of *with highlight*.

Pen Drag&Drop: Continuing the *with highlight* technique, holding the pen button sticks the object to the pen tip. Releasing the button, drops the target at its current location. This is the only *isomorphic* technique for this interaction.

Pen ray pickup: the object snaps to the pen tip after a *pen ray* selection.

Pen Ray Pickup: For the *pen ray* selection there is a depth offset between pen tip and the target. As the user presses and holds the button on the pen, the center of the selected target is snapped to the tip of the pen—similar to [J. H. Lee et al., 2018] but without adjusting the scale of the object to keep its original size. Releasing the button places the target as in *pen drag&drop*.

One-handed & Two-handed Touch Translation: Both the *one-handed* and *two-handed* translation techniques differ only in the hand used to touch the screen—the hand holding the phone for *one-handed*, the hand holding the pen for *two-handed*. Holding the touch attaches the selected object to the phone in its current distance. Changes in position of the phone are directly applied to the position of the virtual object but its rotation in the real world stays the same. Lifting the touch drops the target at its current position. This is similar to the HOMER-S technique by Mossel et al. [2013a].

One-handed & *two-handed*: device movement is applied to the selected object.

Touch Pickup & Pen Drop: This two-step condition combines touchscreen and mid-air pointing device. The user first selects the virtual object by touching and holding the object on the screen. Pressing the button on the pen snaps the object to the tip of the pen as soon as the markers are in view. The touch on the screen can now be released and the object is attached to the pen movement. Releasing the button on the pen drops the target.

Touch&pen: the object is selected by a touch on the screen and then snaps to the pen tip.

6.3.2 Study Setup

Inside the upper half of the interaction volume, we placed 32 cubes with edge length of 3 cm using rules similar to the selection study. Participants had to pick up an object and move it to a virtual drop target shown in one of four possible locations (cf. Figure 6.6). The cube shaped drop targets were shown 5 cm above the corners of the calibration marker with a visual connection to the corner to provide a link to a real world location without being directly tied to a haptic surface. We encourage analyzing interaction techniques that interact with the physical environment. Moving a target inside the drop target, highlighted the drop location. A correctly dropped target moved to its initial position and the next target and drop location was shown. The order of targets and drop locations was randomized.

Users had to pick up a cube and move it to a target close to the table surface.

Figure 6.6: In the translation study, participants had to pick up the highlighted cube and move it to the position of a drop target close to the table.

6.3.3 Study Procedure

Participants were allowed to use *clutching*.

The procedure varied from the selection study only in the task and techniques evaluated. We told the participant that she was allowed to drop and re-select a target as often as she wanted to. This was explicitly shown for *one-handed* and *two-handed* as pilot studies showed a *clutching*-style of multiple drag and drop actions to get a target to a comfortable position relative to the phone. After each condition, we asked the participant to rate the technique for *ease of interaction* and *confidence of interaction*. Overall, each participant moved 160 targets (32 cubes$\times$5 *translation techniques*). The order of conditions was counterbalanced between participants using a Latin square.

6.3.4 Measurements

We measured time, pickups & misses, and subjective ratings.

We measured the time needed to select and drop each target (*task time*). This time was split into *selection time*, measuring the time until the target was picked up, and *translation time*, recording the total duration the target was moved. If the participant placed the target outside the drop location, the *selection time* increased again. Since we had observed a *clutching* technique for *one-handed* and *two-handed*, we recorded how often the participants picked up a target

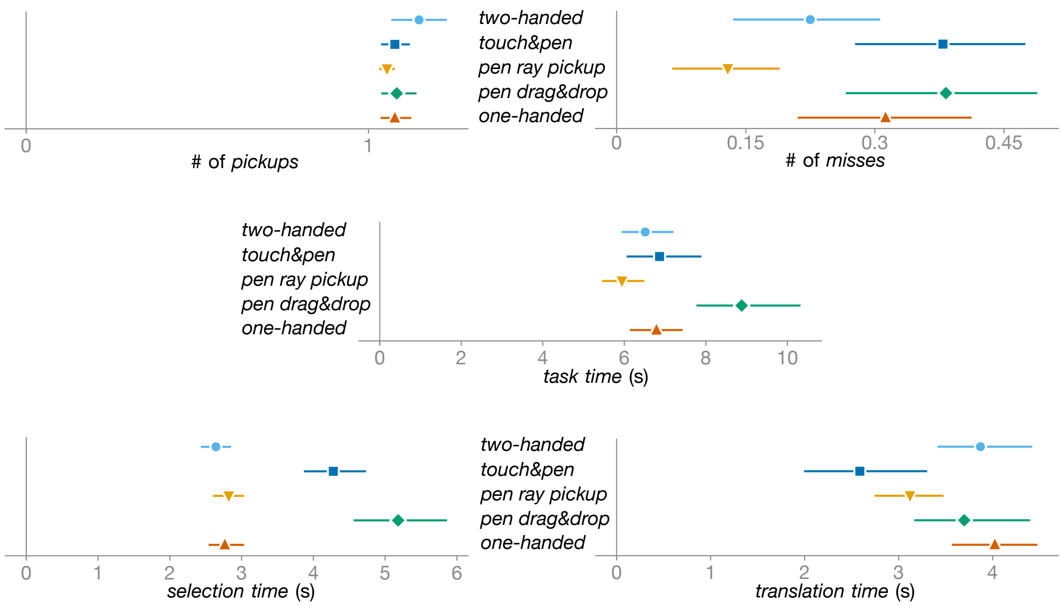


Figure 6.7: Results of the translation study. Means and 95 % CIs without within-subject adjustment.

during a trial (*pickups*). However, this also records unintended drops. Furthermore, we recorded how often the participant missed the target during the selection (*misses*). The subjective ratings were gathered on a 7-level Likert-scale for *ease of interaction* and *confidence of interaction* and as a ranking of the five techniques.

6.3.5 Results

We again recruited 15 participants for this study (2 female, 1 n.a., 22–35 years, M = 27.1 years, SD = 3.6 years, one left-handed). As the phone crashed through a *pen drag&drop* condition, we did not gather data for the remaining targets. Overall, we collected 2383 drag&drop interactions. We averaged measurements for each condition among participants before calculating the within-subject differences.

The overall confidence intervals are shown in Figure 6.7. For details on the evaluation method, we refer to Appendix A. Similarly to the selection study (Chapter 6.2), we present the within-subjects mean differences for our measurements

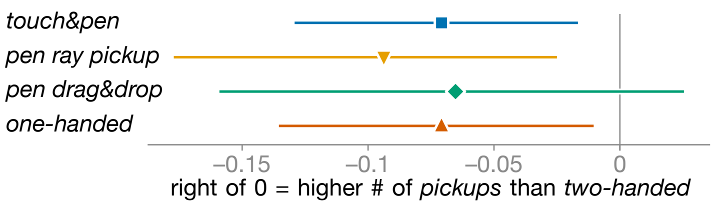
We analyze within-subject differences using confidence intervals.

to compare the performance of using a different *translation technique*. We used *two-handed* as our baseline technique. As before, we did not calculate differences for Likert Scale ratings but report means and bootstrapped confidence intervals. Individual results are calculated and presented the same way. Compared to the findings reported in the published paper [Wacker et al., 2019], this evaluation shows the same major findings but does not show the same minor differences in the subjective ratings.

Number of Pickups

Most users do not pick up a target more than once.

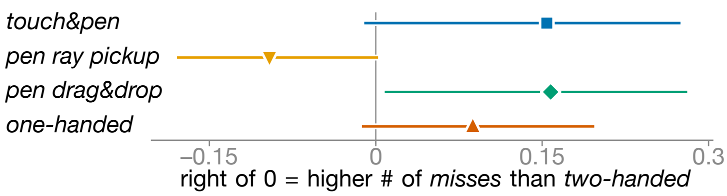
Most users did not pick up a target more than once. The average for all conditions is close to one pickup and there is no great difference between techniques (*two-handed*, M = 1.1 pickups, CI [1.1, 1.2]).



Number of Missed Selections

There are few *misses*.

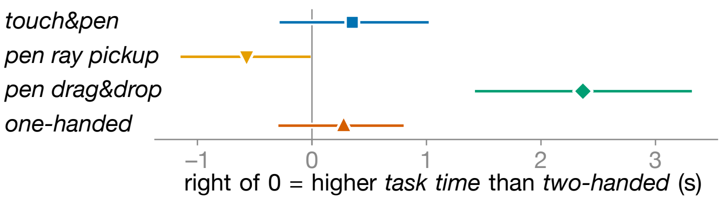
The accuracy of selecting an object was high and there were few *misses* (*two-handed*, M = 0.2 misses, CI [0.1, 0.3]). Similar to the results from the selection study, *pen ray pickup* seems to lead to fewer *misses* but the difference is very low.



Task Time, Selection Time, and Translation Time

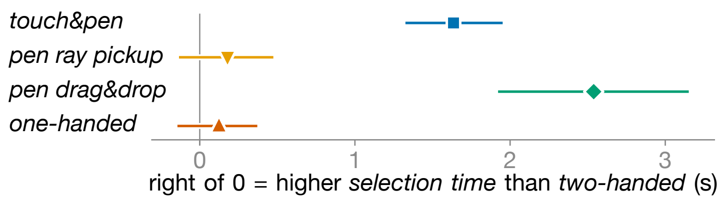
Regarding the overall *task time*, our measurements indicate that *pen drag&drop* requires the longest time overall. The other techniques took a more similar time to complete the task (*two-handed*, M = 6.5 s, CI [5.9 s, 7.2 s]).

Pen drag&drop requires the longest time.



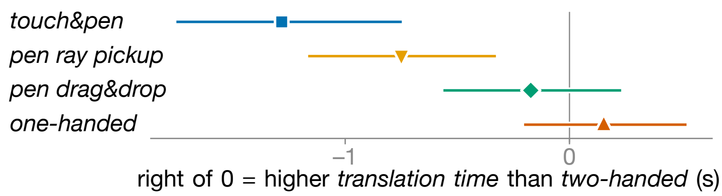
Splitting the task in *selection time* and *translation time* shows for *selection time* durations that are similar to the results from the selection study (*two-handed*, M = 2.6 s, CI [2.4 s, 2.8 s]). However, *touch&pen* seems to take longer even though it also uses ray-casting to select an object.

*Selection time*s are similar to the selection study except for *touch&pen*.



Moving the object to the target seems to take similar amounts of time for *two-handed*, *one-handed*, and *pen drag&drop* (*two-handed*, M = 3.9 s, CI [3.4 s, 4.4 s]). *Pen ray pickup* appears to be faster with *touch&pen* requiring even less translation time.

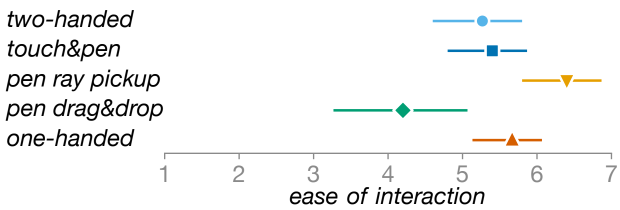
Touch&pen seems to have the fastest *translation time*.

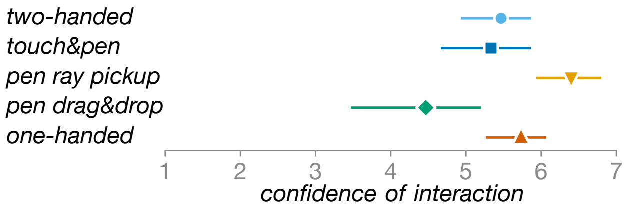


Subjective Ratings

Pen ray pickup seems to be rated higher and *pen drag&drop* lower than the other techniques.

The ratings for *ease of interaction* and *confidence of interaction* are very similar. For both, the conditions are closer together than the ratings in the selection study from the previous chapter. *Pen ray pickup* seems to be rated higher than the other techniques, and *pen drag&drop* seems to be rated lower than *two-handed*, *touch&pen*, and *one-handed*.





The ranking of conditions shows a preference for *pen ray pickup* before *touch&pen*. *Pen drag&drop* was mostly placed last. *One-handed* and *two-handed* are generally placed in third and fourth position (cf. Figure 6.8).

6.3.6 Discussion

Pen drag&drop requires the longest time; *pen ray pickup* is preferred by participants.

The translation techniques performed similar based on *task time* except for *pen drag&drop* which took longer compared to all other techniques because of the increased selection time. Participants also liked this technique the least. *Pen ray pickup* was preferred by the participants and they missed also less targets with this technique indicating, similar to the selection study, that this technique should be used for mid-air selection and pickup.

Even though *touch&pen* needed more time for the selection, the translation time was significantly lower to all other con-

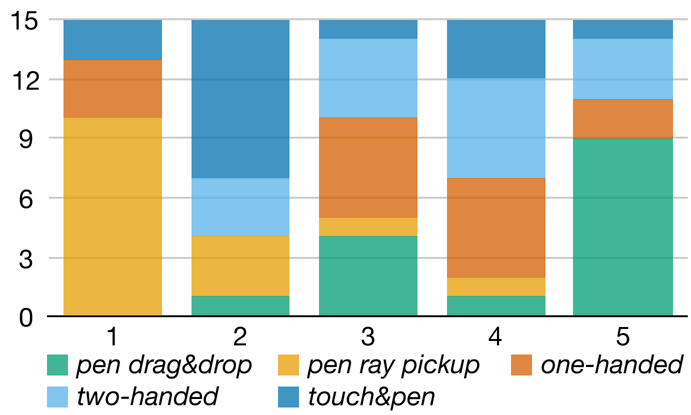


Figure 6.8: Subjective ranking of translation techniques. *Pen ray pickup* is preferred before *touch&pen*. *Pen drag&drop* seems to be least preferred while *one-handed* and *two-handed* are equally distributed at third and fourth place.

ditions and participants ranked this technique only behind *pen ray pickup*. Several participants used a strategy where they left the hand holding the pen on the desk, pressing the button as the other hand touched the object. Moving the viewport to the table snapped the object to the pen which only had to complete a small translation. Participants mentioned that this enabled a more comfortable hand position. As this was only possible because the drop locations were located on the table, further investigation of this technique is necessary to judge its performance for unknown and/or mid-air drop locations. The touch techniques had a similar performance and ranking regardless whether one or two hands were used.

Several participants used a strategy for *touch&pen* that allowed them to rest their hand on the table.

The results indicate that a placement via the pen is beneficial but the pickup needs support to overcome depth issues.

6.4 Rotation & Scaling

Apart from selection and translation, rotation and scaling are other essential manipulation techniques for the interaction with objects in mid-air [LaViola et al., 2017]. The author of this thesis worked together with students to ex-

Rotation and scaling are other basic manipulation techniques.

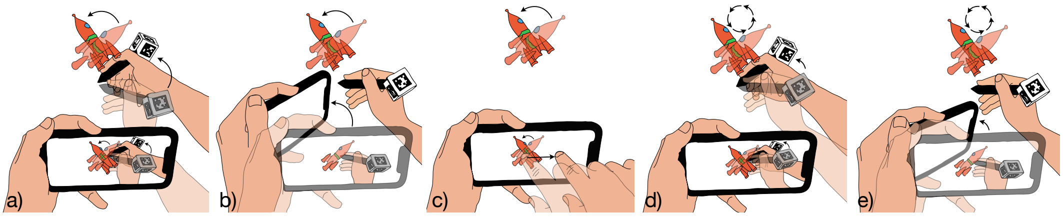


Figure 6.9: Rotation techniques. a) *Direct pen*, b) *Direct device*, c) *Touchscreen*, d) *Pedal pen*, e) *Pedal device*.

We designed techniques and they have been evaluated in students theses.

plore these interactions and the findings have been published in the bachelor's thesis of Donna Klamma for rotation techniques [Klamma, 2019] and in the bachelor's thesis of Farhadiba Mohammed for scaling techniques [Mohammed, 2020]. We designed the interaction techniques together and the author of this thesis provided feedback for the study designs and evaluation.

6.4.1 Rotation Techniques

Using the pose of the device to specify the orientation of an object has shown good results.

Rotation of an object is different from translating an object in that continued rotation in one direction will eventually lead back to the initial orientation. This means that an object can be rotated in many different directions to achieve a particular orientation. Related work on rotation techniques for Handheld AR suggests that using the orientation of the device to specify the orientation of an object in mid-air achieves good results [Henrysson et al., 2007; Harviainen et al., 2009]. Other evaluations used scrolling on the touchscreen to specify the rotation around two axes [Mossel et al., 2013a]. Moving the phone was required to change the available axis to specify the rotation direction more clearly. Systems like the ARPen system add another component to the available interaction modalities: the mid-air pen.

We designed 5 *rotation techniques*.

We designed five *rotation techniques* for the ARPen system, using the three main interaction modalities of the system: touchscreen, mid-air pen, and device movement (cf. Figure 6.9). In order to focus on the rotation techniques, the position of the object does not change while manipulating the orientation of the object.

Direct Pen Rotation: The first technique is using the ARPen to control the rotation. After selecting the object, any rotation of the pen is directly applied to the object as well. This corresponds to sticking a pencil into an apple to control its orientation.

Direct pen: Changes in orientation of the pen are applied to the object.

Direct Device Rotation: Similarly, the orientation of the smartphone can also be used to specify the rotation of the object. Here, the changes in orientation of the device are applied for the object while it is selected. This is similar to the HOMER-S technique by Mossel et al. [2013a], which we also used in the translation study (cf. Chapter 6.3), or the Tilt condition by Henrysson et al. [2007].

Direct device: Changes in orientation of the device are applied to the object.

Touchscreen Rotation: Previous studies evaluating rotation methods have used the touchscreen of the mobile device to adjust the orientation of the selected object. We designed our touchscreen method following the 3DTouch technique from Mossel et al. [2013a]. After an object has been selected, swipes on the touchscreen are applied to the object by mapping the x&y movement to movement around two rotation axes. The orientation of the rotation axis is defined by the viewing orientation of the device. For this, the orientation of the device's local coordinate system is mirrored for the selected object. A dragging movement in x direction on the device is interpreted as a rotation around the y axis and a movement in y direction as a rotation around the x axis. To rotate the object around the remaining axis, the viewing orientation of the device onto the object has to be adjusted.

Touchscreen: Swipes on the touchscreen are interpreted to rotate the object.

***Pedal* Techniques with Pen or Device:** Larger rotations, e.g., to rotate an object around 180 degrees, requires adaption of the interaction. When using both *direct pen* or *direct device*, trying to perform the action in one move requires uncomfortable twisting of either the hand holding the pen or the hand holding the device. One strategy to overcome this issue is *clutching*. Here, the object is picked up, rotated about a comfortable amount, and dropped before returning

Pedal pen & *pedal device*: Changes in the orientation of the pen or the device control the direction and velocity with which the object rotates.

the hand or device to the initial position to pick the object up again [LaViola et al., 2017, p. 261]. Therefore, the one rotation action is separated into multiple smaller rotation actions. In the case of *direct device* this also addresses the issue that the object to rotate is no longer visible on the screen if it is turned away too far. Another potential solution for this issue is to adjust the techniques so that they are not controlling the orientation of the object directly but the velocity by which the object rotates. The direction of the rotation is specified by the direction in which the pen or device has been moved and the size of the angle specifies the speed of the rotation. We named these techniques *pedal* techniques in reference to pedals in cars to specify the velocity of the vehicle. While the *pedal* techniques represent a time-based interaction to adjust the rotation, they remove the need for clutching strategies and visibility issues when using *direct device*. Further evaluation of these techniques is required to judge how the advantages and disadvantages compare against each other.

Pedal techniques remove the need for clutching.

All techniques require the selection of the object that needs to be rotated. In our selection study (Chapter 6.2), both *pen ray* and *two-handed* achieved comparable results and were preferred by participants. Therefore, we decided that the selection for rotation techniques using the pen or device use the *pen ray* technique while the touchscreen technique uses the *two-handed* technique.

Pen ray and *two-handed* are used to select the object.

Klamma [2019] conducted a study to compare these rotation techniques against each other and measured performance metrics as well as subjective preference. Participants had to use the techniques to rotate the model of a rocket ship to match with a less opaque copy of the rocket ship and confirm their final orientation. The ranking of the techniques, which Klamma [2019] collected, indicates that participants liked the *direct* techniques the most and *pedal* techniques the least (cf. Figure 6.10). Participants also picked up objects more often in the *pedal* techniques, suggesting that controlling the velocity is more problematic and requires more frequent stops compared to the *clutching* in the *direct* techniques. Therefore, we suggest to not further consider *pedal* techniques in future rotation studies. While evaluating the results, we found that the implementation of the

Participants had to rotate a model to match with another object.

The ranking of techniques indicates that users like *direct* techniques more than *pedal* techniques.

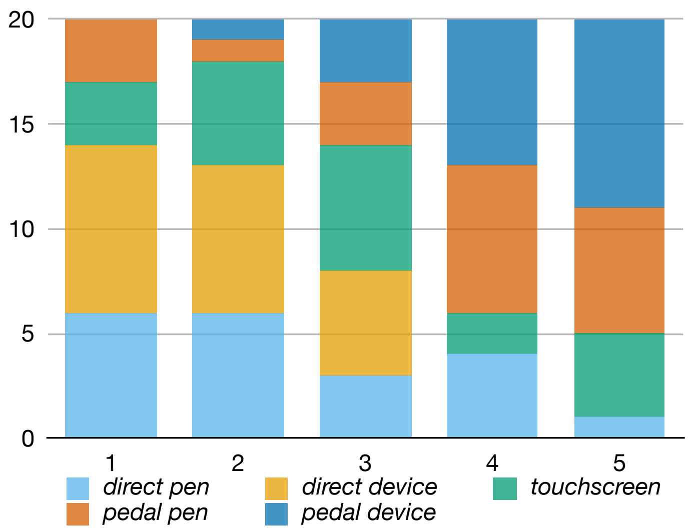


Figure 6.10: Subjective ranking of rotation techniques. *Direct* techniques seem to be preferred to *pedal* techniques. *Touchscreen* was placed on all positions of the ranking.

techniques was affected by a bug causing the calculation of the pen orientation to jitter and even flip occasionally. This strongly influences the *direct pen* technique and, to a lesser extent, the *pedal pen* technique. Also, the speed of rotation in the *pedal* techniques was not controlled gradually dependent on the angle difference but by angle ranges causing jumps in rotation speeds and ranges in which no velocity changes occur even though the angle changes. Therefore, further studies should be conducted to obtain detailed performance measurements. Future studies should also include *non-isomorphic* interaction techniques that do not apply the pen or device rotation one-to-one to the virtual object but instead, apply the rotation with an *amplification factor*. These techniques have shown promising results in VR settings, especially for larger rotations [Poupyrev et al., 2000; Laviola and Katzourin, 2007], even when combined with a simultaneous translation task [Gao et al., 2020].

Calculation of the pen tip was affected by a bug: further studies needed for performance measurements.

6.4.2 Scaling Techniques

The scaling of objects differs from translation and rotation in that for most of the real objects in our environment, it is

Scaling does not have a "natural" technique.

not possible to adjust their size. This means that there does not exist a "natural" scaling operation comparable to the "pick up, move, and rotate" action for translation and rotation tasks. However, on touchscreen devices, the *pinching* gesture has established itself as the standard scaling interaction [e.g., Cohé and Hachet, 2012].

Scaling means to change the size of an object with a *scaling factor*.

Applying the same scaling factor to all dimensions is called *uniform scaling*.

Scaling an object means to change at least one of the width, height, or depth values of the object. To show the individual dimensions, we used an axis-aligned bounding box around the object—the smallest cuboid containing the object for which the axes are aligned with the world-coordinate axes. This allows for an easy representation of the size in each dimension regardless of the precise geometric shape of the object (cf. Figure 6.11 for an example). The amount of scaling is specified by the *scaling factor*. If the same scaling factor is applied to all dimensions (width, height, and depth), the operation is called *uniform scaling* as the proportions of the object are unchanged, compared to a stretching or compressing if only one or two dimensions are scaled. For the design and evaluation of scaling techniques, we decided to focus on uniform scaling. However, all of our interaction techniques can also be used to specify the scaling factor along one or two dimensions. We provide a discussion on how to adapt the interaction techniques.

The position of either the center or a corner is fixed during the scaling action.

Another decision for scaling actions is the direction in which the size change is applied. There are two main possibilities for this. First, the position of the object stays fixed during the scaling. This means that the extent of the object changes in all directions (cf. Figure 6.11, left). An example of a scaling like this is the pinch-to-zoom method on mobile devices. The second possibility is to keep a corner or side static and change the extent of the object only in the opposite directions. This type of scaling happens, for example, when adjusting the size of a window in desktop operating systems: dragging a corner of a window does not adjust the position of the opposite corner (cf. Figure 6.11, right). For the design of scaling techniques with the application context of *situated modeling*, we decided to focus on the second option of scaling into a specific direction as this allows placing an object on a surface and then specify the size without the object "breaking" through the surface.

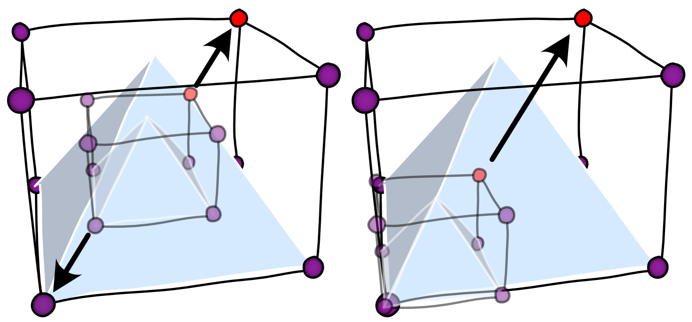


Figure 6.11: *Left*: Scaling without changing the center position of the object. The object expands in all directions when the size is increased. *Right*: Scaling by fixing a corner. The position of the corner does not change during the scaling operation.

For our scaling techniques, we therefore had the following design decisions:

Design decisions.

- a corner of the bounding box needs to be clear as the fixed corner that does not change during scaling,
- the same scaling factor is applied to all dimensions equally.

Following, we describe the six *scaling techniques* that we designed for the ARPen system (cf. Figure 6.12).

We designed 6 *scaling techniques*.

Pinch Gesture Scaling: We included the *pinch* technique as it is currently used in many touchscreen scenarios and related work shows that users like this technique. To specify the fixed corner, the user first has to select the corner of the bounding box that should "move" while scaling, using the *two-handed* selection technique. Afterward, the scaling factor is calculated by dividing the current distance between the pinching fingers by their starting distance.

Pinch: The scaling factor is specified by pinching on the screen.

Scroll Gesture Scaling: The *scrolling* technique mimics the scaling behavior of windows on desktop systems. The user can touch and drag a corner of the bounding box to

Scrolling: Dragging a corner on the screen scales the object.

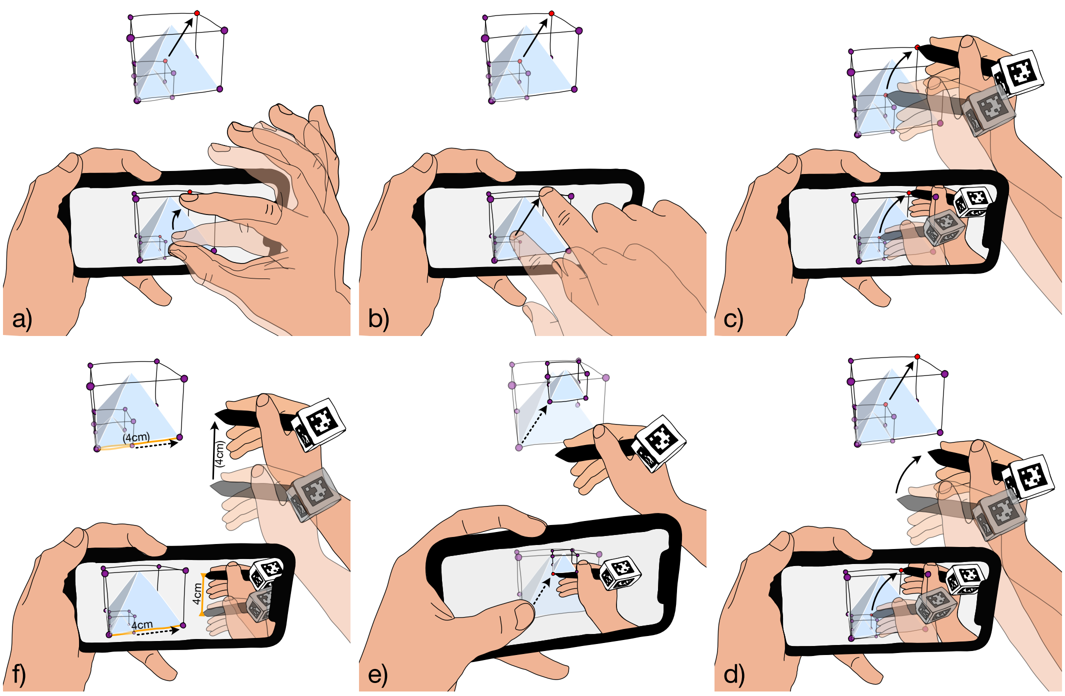


Figure 6.12: Scaling techniques. a) *Pinch*, b) *Scrolling*, c) *Direct pen*, d) *Pen ray*, e) *Touch&pen*, f) *Distance*.

scale the object. The opposing corner is then assumed as the fixed corner. The moving corner always stays on the diagonal of the object to result in a uniform scaling. If the user is not dragging along the projection of this diagonal, the corner moves to the closest point on the diagonal.

Direct pen: Users have to move the pen inside a corner and drag it.

Direct Pen Scaling: For *direct pen*, the interaction is similar to *scrolling* but in mid-air and controlled with the pen. The user has to grab a corner by moving the pen tip inside the corresponding sphere of the bounding box. Pressing the button on the pen allows moving this corner around and the opposite corner is taken as the fixed corner. The moving corner always stays on the diagonal of the object to allow for uniform scaling. If the user is deviating from this diagonal, the sphere moves to the closest point from the pen tip to the diagonal. Releasing the pen button, stops the scaling at the current size.

Pen Ray Scaling: This technique adapts the *direct pen* technique in that it is possible to select the corner to move by the *pen ray* selection technique. Again, the opposite corner is taken as the fixed corner. The scaling factor is calculated by interpreting the movements after selection relative to the selected corner. This means that the scaling operation looks similar to the *direct pen* technique if the original viewing position does not change. However, if the user moves the viewport, the difference between pen position and corner becomes more visible.

Pen ray: The corner is selected by *pen ray*. Movement is then applied to the corner.

Scaling with Touch & Pen: Previous results in the translation study (Chapter 6.3) have indicated that combining the touchscreen and pen interaction can be beneficial for the interaction. The *touch&pen* technique combines the touchscreen selection of a corner with the position information of the pen. The user taps and holds the corner that should be moved for the scaling operation. Pressing the button on the pen moves this corner to the closest position on the diagonal in respect to the pen tip position. Further movement of the pen can adjust the scaling and releasing the button confirms the current scale.

Touch&pen: Users select a corner on the screen and it snaps to the pen position.

Distance Scaling: In all previous techniques, the scaling operation continuously adapts the size of the object. However, in some situations it might be helpful to specify the exact size of one dimension, e.g., so that the scaled object fits inside an existing space. An example from the area of *situated modeling* is to size the model of a cup to fit inside an existing cup holder. We designed the *distance* technique to allow for such tasks. Instead of a corner, the user selects an edge of the bounding box to indicate which dimension she wants to specify. In a second step, the user draws a line by specifying a start and an end point with button presses. The distance between these points is then taken as the target length for the selected edge of the bounding box. This means that the scaling factor is defined by the ratio between the current length of the edge and the distance of the line.

Distance: Users select an edge and then drag a line to specify the length the edge should have.

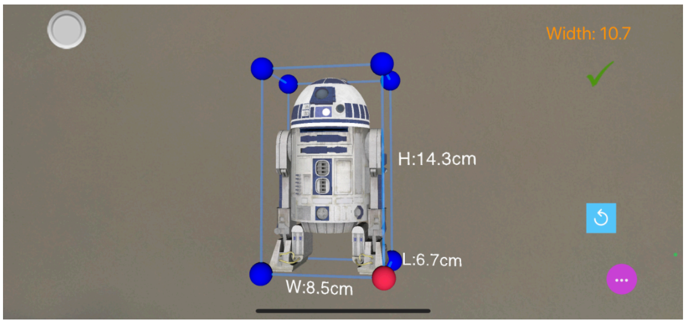


Figure 6.13: Setup of the scaling task. The lengths of edges adjacent to the selected corner were displayed. Participants had to scale the object so that it matches the value of the dimension specified in the top right.

Evaluation of Scaling Techniques

Users had to scale an object to match a given dimension.

Mohammed [2020] implemented our designed techniques for the ARPen system and evaluated them in a user study with 24 participants. In her study, participants had to use the techniques to scale objects to match a given size. The target size was given by a label on the screen telling the participant the target dimension (width, height, or depth) and target value. During the scaling action, the current lengths of the edges adjacent to the moving corner were displayed (cf. Figure 6.13). Participants had to confirm their current scaling size by pressing a checkmark button on the screen.

Mohammed measured time, deviation, scaling attempts, and subjective ratings.

Mohammed measured the time it took participants to complete the scaling task (*interaction time*), the deviation between the final and target size (*deviation*), and the number of scaling attempts per trial (*attempts*). Participants were also asked for *subjective ratings* to assess the perceived ease of use, precision, and complexity as well as how stressful the interaction was. After the participants had used all techniques, she also asked them for a *ranking* of the different techniques from one to six.

Distance & *touch&pen* required more time.

For the *interaction time*, Mohammed found that using *distance* took significantly more time (M = 22.26 s, SD = 10.49 s) than all other techniques. *Touch&pen* (M = 15.74 s, SD = 7.05) also required a significantly longer interaction compared to the other techniques, which are closer together (*scrolling*: M = 9.47 s, SD = 3.65 s; *pinch*: M = 9.85 s, SD = 3.26 s; *pen ray*: M = 11.31 s, SD = 3.58 s; *direct pen*: M = 12.17 s, SD = 6.56 s) with a trend that the touchscreen methods are a bit faster than the pen-based methods. As participants could refine the scaling results before confirming their final size, the *deviation* was minimal for all the techniques (less than 1 mm on average). Regarding the scale *attempts*, many trials required only one scaling interaction. *Touch&pen* required significantly more attempts (M = 4.42 attempts, SD = 2.39 attempts) compared to the other techniques. *Pen ray* (M = 2.79 attempts, SD = 1.67 attempts) took significantly more attempts compared to *scrolling* (M = 1.33 attempts, SD = 0.48 attempts) and *direct pen* (M = 1.62 attempts, SD = 0.71 attempts). The remaining techniques did not show additional significant differences in scaling attempts (*pinch*: M = 2.04 attempts, SD = 1.08 attempts; *distance*: M = 2.46 attempts, SD = 1.59 attempts).

Deviation was minimal for all techniques.
Many trials required only one scale attempt.

For the *subjective ratings*, Mohammed found that participant rated *pinch*, *scrolling*, and *pen ray* better than *direct pen* and *distance*. The ratings for the *touch&pen* technique are mostly between these two groups. While the measured deviation did not show any significant differences, participants reported a significantly higher precision for *pinch* compared to *distance* and *direct pen*. The other techniques were not rated significantly different regarding precision. Finally, this is also apparent in the *ranking* of the techniques (cf. Figure 6.14). It is clear that *direct pen* and *distance* were placed on lower places more frequently. However, the top side of the ranking is less clear. *Pinch*, *scrolling*, *pen ray*, and *touch&pen* are often placed on the first places with a slight preference for *pinch* and *pen ray*. Among the touchscreen methods, *pinch* seems to be more preferred while for pen-based methods, *pen ray* seems to be more preferred.

Participants rated *pinch*, *scrolling*, & *pen ray* higher than *direct pen* & *distance*.

Mohammed also mentions qualitative feedback participants gave during the study. For *direct pen*, participants mentioned that the hardest part is finding and selecting the corner and that this would be more relaxing in the *pen ray* technique. For *touch&pen*, participants said that this technique allows for larger immediate scaling steps by not having to pick up a corner. On the other hand, other participants used this technique similar to *pen ray* and felt that

Users reported less issues with *pen ray* than with other pen techniques.

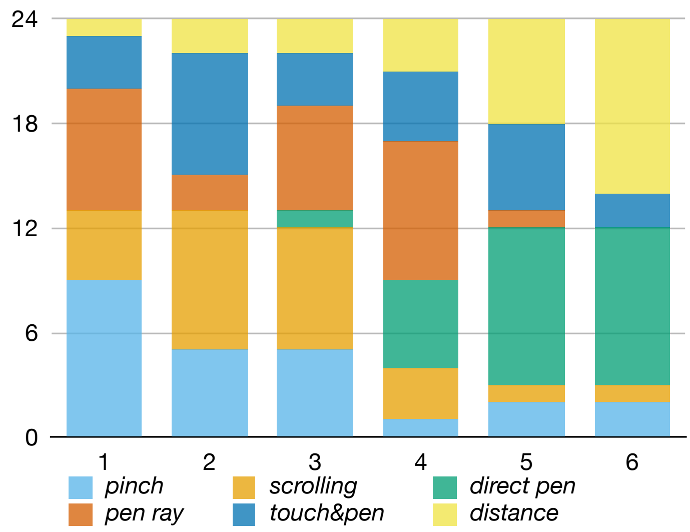


Figure 6.14: Subjective ranking of scaling techniques. Both *distance* and *direct pen* are mostly placed on the last places. *Pinch* and *pen ray* seem to be slightly preferred to *scrolling* and *touch&pen*.

Participants mentioned that they felt more in control with touchscreen methods.

the touchscreen selection was redundant. For the *distance* technique, participants mentioned that it was harder to understand. Also, the participants noted issues drawing the line if the target scale was very large, requiring adjustment of the viewport. Furthermore, participants noted that it is necessary to redraw the whole scaling line if the resulting scale does not match the intended scale. Comparing touchscreen-based and pen-based techniques, participants mentioned that they felt more in control with the touchscreen methods but that the limited space and occlusion is a problem. On the other hand, they also mentioned enjoyment when using the pen techniques. However, coordinating the pen with markers and the camera was overwhelming for several participants.

Discussion of Scaling Techniques

Pinch is the preferred touchscreen technique.

The results from the study by Mohammed [2020] indicate that touchscreen-based techniques achieved generally good results, both in the measured data as well as subjective ratings. Between *pinch* and *scrolling*, *pinch* seems to be preferred even though the required selection of a corner introduces another step to the well known pinch-to-zoom action. On the side of pen-based methods, *direct pen* achieved good performance results but did not receive high ratings from the participants. *Pen ray*, on the other hand achieved a lower performance for the scaling *attempts* but was rated high by participants. This indicates that the issues of finding the precise location of a corner to pick up weighs stronger than smaller performance issues when using the pen ray selection and having less direct interaction. However, the cause of the increased scaling *attempts* with *pen ray* are not clear and should be investigated more closely. The *touch&pen* technique required significantly more scaling *attempts* but otherwise did not perform particularly good or bad. Observations during the study indicate that several participants did not use the finer control of placing the corner after it had snapped to the pen tip. Instead, they adjusted the position and pressed the button again, causing the count of another scaling *attempt*. Using a line to specify the new scale in the *distance* technique achieved generally lower scores and also required longer *interaction times*. However, the current study did not evaluate the use-case in which we envision this scaling technique to be suitable: specifying the scale so that the object fits between physical objects in the scene. So while the *distance* technique is not recommended for general mid-air scaling tasks, it should not be left out when investigating these more specific scaling operations.

Pen ray is the preferred pen technique.

Overall, the recorded *interaction times* are often higher than we initially expected. One reason for this could be, that participants tried to be as precise as possible and required longer to "hit" the exact value without jitter of the pen tip or finger changing the value. A future study could, for example, allow a threshold around the target value.

Interaction times are higher than expected.

Adapting the Techniques for Other Scaling Operations: The study by Mohammed [2020] evaluated mid-air scaling tasks with uniform scaling and scaling into one direction. The results provide an initial understanding of the performance of these techniques. Following, we provide suggestions on how the techniques can be adapted to also be

We provide suggestions for non-uniform scaling and scaling in all directions.

used for non-uniform scaling and scaling in all directions. While scaling in all directions is trivial for all techniques and even makes the *pinch* technique simpler, allowing for non-uniform scaling requires additional considerations of how to specify individual scaling factors for each dimension separately using the techniques.

Pen-based techniques can control each dimension individually.

For the pen-based methods *pen ray*, *direct pen*, and *touch&pen* the change is simple as the pen position would simply not have to be projected onto the diagonal of the object but rather interpreted directly to calculate the scaling factors for each dimension. If the same scaling factor is applied to one or two dimensions instead of all three, the necessary change for the *distance* technique is to allow the individual selection of dimensions that should be scaled—selecting all three dimensions results in a uniform scaling while only selecting one or two dimensions results in a non-uniform scaling. However, if different scaling factors should be applied to different dimensions, users would have to scale each dimension individually.

Specifying different scaling factors for touchscreen techniques requires multiple steps.

For the touchscreen techniques, this is similar. If the same scaling factor is applied to two dimensions, simply locking one dimension would be enough to enable non-uniform scaling. However, controlling different scaling factors at the same time is more problematic and would require multiple steps. This is especially true for *pinch* as the pinch gesture only specifies one scaling value through the change in distance between the pinching fingers. For *scrolling*, it is possible to calculate two individual scaling factors from the movement on the touchscreen, for example by projecting the scrolling movement into the plane of one side of the bounding box. However, specifying which plane to use would be necessary and specifying a scaling factor for the third dimension is not directly possible.

Summary & Future Work

The study of mid-air scaling techniques by Mohammed [2020] showed similar results and preferences for touchscreen methods such as *pinch* and *scrolling* as well as using the pen with the *pen ray* technique and we suggest using these techniques for similar situations. This study investigated uniform scaling of the objects and further studies should look at non-uniform scaling. While pen-based techniques would allow for individual control over all scaling factors directly, touchscreen-based techniques would require multiple steps. It will be interesting to see if users are even able to control and perceive changes in all dimensions correctly or if a multi-step interaction controlling each dimension individually is better. Another interesting direction to continue is to look at different scaling scenarios and tasks. In the presented study, numerical targets were given for a specific dimension. Future projects could compare if scaling not based on numerical values but other aspects such as other objects in the scene, results in different findings. A specific subset of this is the inclusion of physical objects that should be used to specify the target size such as for a modeling task. In that case, it will be particularly interesting to see if the *distance* technique performs better compared to this study and if the improvement warrants its use for such a specific scenario alone.

This study investigated uniform scaling techniques and found that *pinch*, *scrolling*, & *pen ray* achieve best results.

Other scaling scenarios are an interesting direction.

6.5 Manipulation Techniques: Summary & Future Work

Our studies with the ARPen system show that the manipulation of objects in mid-air can be achieved with a variety of techniques in Handheld AR with a mid-air pen. Combined with the geometry creation capabilities of the ARPen system, having these interaction techniques enables users to create objects such as the cup holder inset by creating basic elements, arranging them using these *manipulation* techniques, before combining them (cf. Figure 6.15).

Adding *selection & manipulation* techniques to the ARPen enables *situated modeling*.

For the selection of virtual objects, we have seen that an *isomorphic* selection in mid-air is not optimal as it takes longer to find the correct 3D position and that it is not possible without highlighting or pen tip visualization. Here, *non-isomorphic* techniques using ray-casting improve the interaction. However, placement by moving the pen to the in-

Ray-casting techniques achieve good selection results.

Figure 6.15: With the ARPen system, the user can specify volumetric objects by tracing over physical objects (*left*). With our *selection & manipulation* techniques, the objects can be arranged relative to each other (*right*) to be combined into more complex objects.

Users prefer the interaction by casting a ray through the tip of the pen.

tended position ranked high on user preference. Over the selection and translation studies, users preferred the selection and pickup via a ray shot through the pen tip. The 2D representation of the scene allowed aligning the pen tip with the object to select it. This interaction also achieved the highest amount of successful selections, short selection and translation times comparable to touch techniques, and least device movement based on the size of the projected targets. We recommend employing ray-based techniques for manipulation tasks in Handheld AR modeling systems.

Touch techniques perform well for selection and scaling techniques in particular.

Touch techniques, especially with two hands, achieve good results particularly for selection and scaling techniques. For selection, *two-handed* is comparable to *pen ray* in user preference followed by *one-handed*. However, the high projected size of the targets suggests that users have to move the device more through the scene. For the scaling of virtual objects, both *pinch* and *scrolling* showed good results. As the scaling study was focussed on precision, this could mean that the added support from the touchscreen positively affected the interaction.

The combination of touchscreen and mid-air pen has potential.

For the translation and scaling of objects, *touch&pen* shows that the combination of touchscreen and mid-air pointing device has potential. The possibility to select an object mid-air via touch and then place it near the table without having to continuously lift the pen, allowed users to rest their arm providing a more ergonomic interaction. This is particularly important since users mentioned fatigue in their arms during the studies. For the scaling study, *touch&pen* also showed good results but participants seem to favor the touchscreen techniques or *pen ray* technique.

Direct rotation techniques are preferred to *pedal* techniques.

Participants preferred to use *direct* rotation techniques, that applied the rotation of the pen or device to the virtual object, to *pedal* techniques, that control the velocity of rotation. Comparing these *direct* techniques to *non-isomorphic* techniques, seems like a promising direction for future studies of rotation techniques.

A next step could be developing a consistent set of interaction techniques.

The next step in the area of *selection & manipulation* in Handheld AR with a mid-air pen could be the development of a consistent set of interaction techniques. In this chapter, we have investigated each basic action individually but especially translation and rotation are often used together. Future work could design compound interactions that allow the user to specify multiple basic operations simultaneously, and compare how they perform against using the "best" individual technique but having to switch modes.

Investigating how precision requirements affect the interaction is an interesting direction.

The scaling study in particular required precise input since an exact value was given as the target. Potentially, this led to an increased preference for touchscreen methods that offer the support of a physical surface. Further studies could look at this effect more closely to determine whether a similar shift happens for other interaction techniques. Having the option to draw on the touchscreen as in [Arora et al., 2018] or even adding haptic feedback such as vibrations when contacting virtual objects could improve the interaction as well.

Studies outside the lab could help to increase external validity.

Our evaluations so far have been lab studies to control outside effects and focus on the performance using different interaction techniques. Studying the use of *selection & manipulation* techniques in a more natural setting will increase the external validity. Especially situations in which physical surfaces and objects are involved, provide interesting research opportunities. For example, specialized techniques such as the *distance* scaling technique might perform differently in such a setting.

In the next chapter, we shift our focus from *selection & manipulation* techniques to *system control* tasks, specifically graphical menus.

Chapter 7

Context Menus in Handheld AR with a Mid-Air Pen

The *selection & manipulation* techniques presented in the previous chapter represent essential interactions with virtual objects in Handheld AR with a mid-air pen. However, another category of central interaction tasks in 3D user interfaces is *system* (or *application*) *control* [LaViola et al., 2017; Dachselt and Hübner, 2007]. An example of tasks in this category are menu interactions. Menus are an essential component in most visual interfaces and they can take many different forms. One kind of menu is the context menu [Dachselt and Hübner, 2007] which is often used to adjust the properties of a specific object. In the context of an Augmented Reality modeling application, such properties could, for example, be the color or transparency of an object in the scene. We asked ourselves the question how context menus can be realized for a Handheld AR system

Context menus are an example for *system control*.

Publications: The work presented in this chapter has been done in collaboration with Oliver Nowak, Simon Voelker, and Jan Borchers. It is published as a full paper at ACM MobileHCI '20 [Wacker et al., 2020a]. The author of this dissertation developed the research idea and is the main author of the paper. He designed the techniques as well as planned and executed the study together with Oliver Nowak. The author also analyzed all data from the study. Most sections in this chapter are taken from the full paper publication.

A system like the ARPen offers different interaction methods.

with a mid-air pointing device. Such a system offers different interaction methods that could be used to interact with menus. Besides using the touchscreen, the mid-air pointing device can be used for interaction. Also, the device position and orientation in space might be beneficial to use as it would not require moving the mid-air pen. Since the physical world can improve the interaction in mid-air systems (cf. Chapter 3), a menu on a physical surface in the scene might also improve menu interaction. The different interaction methods can all have different advantages and disadvantages for the interaction with menus.

We designed and evaluated different menu techniques.

We implemented basic menus for the interaction methods and evaluated them in a user study. We compared their performance and collected subjective preferences. Based on the results of our study, we consider advantages and disadvantages of each interaction method to assist interaction designers in picking suitable menus for different contexts.

In summary, our contributions are:

- an empirical study about menus using different interaction methods of Handheld AR systems,
- design considerations for such menus based on their advantages and disadvantages.

In the next section, we provide an overview of the related work in the area menu interaction with a focus on VR and AR environments. Afterward, we explain the different menus we used in our study and present the study design and procedure, before discussing the results.

7.1 Related Work

Most studies focus on *selection & manipulation* rather than *system control*.

For most of the projects previously mentioned in this thesis, users interact with the system using one interaction method, and the focus of studies is mostly on *selection & manipulation* rather than *system control* tasks. So far no study has investigated the effect of different interaction methods on the performance of *system control* tasks in the form of menus. In our study, we used a Handheld AR system with a mid-air pen to evaluate the performance of basic menus using the different interaction methods.

Pie menus achieved good results.

Early HCI studies suggest that pie menus are faster compared to linear menus [Callahan et al., 1988; Das and Borst, 2010]. With regard to the number of elements in a menu, studies have found that eight items per level achieves good results and this number is used in several menu evaluations [Kiger, 1984; Landauer and Nachbar, 1985; Miller, 1981].

Menus in VR & AR have been studied since the 90s.

Menus in VR and AR have been the subject of research for a long time [Feiner et al., 1993; Jacoby and Ellis, 1992; van Teylingen et al., 1997]. One of the earliest projects is by Jacoby and Ellis [1992], who placed linear menus as floating windows in a VR environment and interacted with them via hand-pointing gestures. Feiner et al. [1993] were one of the first to implement menus for AR contexts. They discern the placement and attachment of menus in the scene between *object-stabilized*, *screen-stabilized*, and *world-stabilized* windows. This differentiation has also been picked up in other projects [e.g., Dachselt and Hübner, 2007; Kim et al., 2000; H. Lee et al., 2011] (see Chapter 2.2.2). For Virtual Reality, Das and Borst [2010] evaluated various menu setups and interactions. They compared ray-casting to techniques that attach a cursor to the menu and control it indirectly (PAM: pointer-attached-to-menu). Furthermore, they compared contextual menus around a specific object against menus that are fixed in the world. Their results indicate that ray-casting control is generally faster compared to indirect setups and users prefer ray-casting techniques. The contextual menus around a specific object were used faster than world-fixed menus. However, the authors did not detect differences in error rates between these menu styles [Das and Borst, 2010]. Dang and Mestre [2011] found out that the performance of menus also depends on their orientation in space and that accuracy on horizontal menus decreases compared to more angled menus. H. Lee et al. [2011] developed a system for head-mounted AR in which a smartphone is used as a controller to interact with menus. However, they did not perform a user study to compare the different interactions.

Object-stabilized menus were faster than *world-stabilized* menus.

Dachselt and Hübner present a taxonomy for menus in 3D environments.

Dachselt and Hübner [2007] present a good overview and a taxonomy for menus in virtual environments. The central elements that they use to classify menus are *intention of use* (e.g., number of items), *appearance and structure* (e.g., structural layout), *placement* (e.g., frame of reference), *invocation and availability* (e.g., visibility), *interaction and I/O setting* (e.g., interaction device), *usability* (e.g., evaluation criteria), and *combinability*. The menu techniques evaluated in this work occupy similar spaces in this taxonomy but differ on aspects defined by the different interaction methods.

In this chapter, we present and evaluate different menu techniques for a Handheld AR system with a mid-air pen. Such a system presents multiple methods that users can use to interact with the system and each might provide different advantages and disadvantages for *system control* tasks such as menus.

7.2 Menu Techniques

We based our menu techniques on the *methods of interaction* in Handheld AR with a mid-air pen.

In this section, we describe the menu techniques we designed and implemented for our study. We based our designs on the main *methods of interaction* that are directly available to a Handheld AR system with a mid-air pointing device since they are the necessary hard- & software capabilities for such a system to work.

1. Touchscreen.

The first *method* is the interaction with the touchscreen of the smartphone. The screen is needed to show virtual objects in the scene and enable the user to interact with them. To interact with virtual objects, a ray is cast into the scene to decide which object is being targeted (cf. Chapter 6.2). The touchscreen can also be used to show static elements at fixed positions on the screen. The mid-air pen is another essential part of such systems. Therefore, the next *method* is the interaction with this pen. This interaction can either be mid-air—used, for example, to draw lines in mid-air—or by tracing over physical surfaces in the scene. These surfaces have to be tracked by the system to calculate the position of the phone in the scene (cf. Chapter 4). We distinguish between these methods that use the pen, since interacting

2. Mid-air pen.

with physical surfaces can aid the precision of input in immersive environments (cf. Chapter 3) but it is restricted in its availability while mid-air interaction can be in-place all the time. The third *method* to interact with the system is through the movement of the viewing device itself. Adjusting the viewport into the virtual world is an essential part of Handheld AR and the changes of orientation and position of the device can be used as input parameters. Other methods of interaction, such as back-of-device or voice interaction, add additional layers to the interaction with a Handheld AR system with a mid-air pen. However, they are not required for the main functionality of the system and we focused on the main interaction methods instead.

3. Device pose.

Interacting with a menu requires at least two steps. The first step is to open the menu. For a context menu, this means that the user has to select the object to open the menu for (*object selection*). The second step is to select an item from the menu (*item selection*). Since opening the menu is a standard object selection task, we used the two most promising techniques from our selection study (Chapter 6.2) for this step. In that study, the ray-casting techniques *two-handed touch selection* and *pen ray selection* performed with similar success and preference. For *two-handed touch selection* the ray is cast from the location of the touch on the screen and for *pen ray selection*, the ray is shot from the camera through the pen tip into the scene. The first target hit is returned as the selection. For the *object selection* step of our menu techniques, we picked a selection technique based on the main interaction method and our use case scenario of *situated modeling*. Only for our baseline condition of the two-handed touch menu, we decided to use the *two-handed touch selection* technique. Since an application can require successive menu interactions before returning to mid-air input, for example setting colors for different objects, we did not want to put the two-handed touch menu at a disadvantage by using an *object selection* technique that requires additional movement instead of a technique that performs similarly well. The remaining menu techniques either use the pen for interaction, making the *pen ray selection* technique the preferred choice, or have no clear choice of *object selection* technique, in which case *pen ray selection* allows the user to hold a 3D position while interacting with the menu.

Interacting with a menu requires at least two steps.

We use *pen ray* and *two-handed* for *object selection*.

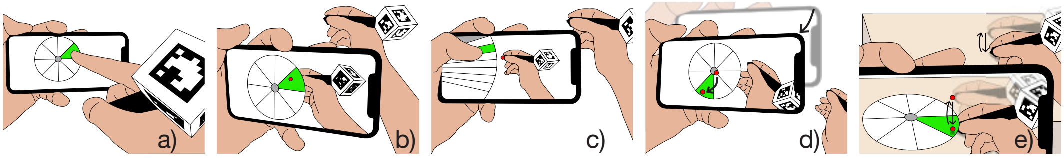


Figure 7.1: The menu techniques used in our study. a) *two-handed touch* as the baseline, b) *mid-air pen*, c) *one-handed touch*, d) *device pointer*, e) *surface*.

We use different methods for *item selection*.

The second step of selecting an item from the menu is different from selecting an object in that it is not necessarily tied to the location of the target anymore. This enables the possibility to present the menu in different locations and also to interact with it using any of the different interaction methods, which might lead to different results compared to our previous mid-air selection study (Chapter 6.2). Table 7.1 shows the combinations of *object selection* and *item selection* for each menu technique.

Menu Technique	Object Selection	Item Selection
Two-handed touch	Two-Handed	Two-Handed
Mid-air pen	Pen-Ray	Pen-Ray
One-handed touch	Pen-Ray	One-Handed
Device pointer	Pen-Ray	Device Motion
Surface	Pen-Ray	Surface Tap

Table 7.1: Combination of object and item selection technique for each menu technique

We use radial menus in mid-air.

We decided to use radial menus for all our mid-air menus as they have shown best results in the literature [e.g., Callahan et al., 1988; Das and Borst, 2010; Komerska and Ware, 2004] and keeping them the same reduces the impact of different presentations. In the center of the menu, we placed a 'back' button allowing the user to close the menu.

We developed 5 *menu techniques*.

In the following, we describe the five *menu techniques* we used in our study (cf. Figure 7.1).

Two-handed touch: *object* & *item selection* by touch.

Two-Handed Touch Menu: The two handed touch menu is modeled after the most often used way of interacting with Augmented Reality content. The phone is held in the

Figure 7.2: The *mid-air pen* menu opened as a circular menu at the target location. The pen tip was always rendered above the menu to be used as a cursor.

non-dominant hand while the dominant hand interacts on the screen. The menu is opened by tapping on the rendering of the virtual object on the screen. A radial menu with roughly 90 % of the height of the screen then opens around the selected target in mid-air and the user can select an item by tapping on it. Once the finger touches the screen, the item underneath it is highlighted. The menu always faces the camera when the phone is moved around the object. Tapping outside of the menu or on the inner area closes the menu without a selection.

Mid-air pen: *object* & *item selection* by *pen ray*.

Mid-Air Pen Menu: The mid-air pen menu uses the same visualization as the two-handed touch menu. However, opening the menu and selecting an item is done with the pen-ray technique. While the menu is open, only the tip of the pen is visualized in front of the menu. This allows using the pen tip as a cursor while avoiding occlusion of items by the user's hand (cf. Figure 7.2). Hovering over an item highlights it.

The action after pressing a button depends on the element behind the cursor.

For this and the remaining menu techniques, the action after pressing a button on the pen while the menu is open depends on the element behind the visible cursor. If the cursor is over a menu item, this item is selected. If the cursor is beside the menu, the menu closes, which is a similar behavior to clicking outside of an open menu in current operating systems.

One-handed: The menu is shown on the left side of the screen.

One-Handed Touch Menu: The one-handed touch menu is similar to a normal table view that is shown on the left side of the touchscreen. After the menu is triggered by the mid-air selection, it is opened to the left of the touchscreen so that an item can be selected with the hand holding the phone. We chose the size of the menu based on the results of our grasp-study in Chapter 4.3. The shape of the menu follows the rotational movement of the thumb making it easy to touch each element. The top space of the menu is used for the 'back' button and pressing any button on the mid-air pen closes it as well.

Device pointer: Device movement controls the cursor.

Device Pointer: The device pointer uses the movement of the device to select an item in the menu. The menu is opened with a pen-ray selection and once the menu is open, a cursor is shown in the center of the menu. This cursor behaves as if a ray is cast from the camera onto the menu, which means that the translation and rotation of the smartphone is applied to the cursor. The cursor is also clamped to the size of the menu so that no overshoots are possible. If the user moves the device further than the edge of the menu, she has to return to the edge before the cursor moves again. This corresponds to an absolute mapping of the device movement onto the cursor movement. The item underneath the cursor is highlighted and pressing a button on the pen selects the item. While the cursor of this technique is shown, the sphere on the tip of the pen is removed to avoid confusion.

Surface: The menu is shown aligned with the table. *Item selection* by tapping the surface.

Surface Menu: The surface menu takes the knowledge about the environment into account. After a menu is triggered by a pen-ray selection, the pie menu is shown on the table surface in front of the user. Pie menus have also shown good results in tabletop scenarios [Komerska and Ware, 2004], so we decided upon this structure for the menu to keep the differences in visualization minimal compared to the other menu techniques. As the user moves the pen close to the menu on the table, the object she is hovering over gets highlighted. Tapping the surface and lifting the pen selects the tapped item. Since depth perception is a known problem of Handheld AR (cf. Chapter 2.3), we display the shadow of the pen tip on the menu to aid the user in determining over which item they are currently hovering. The other techniques work by using ray-casting techniques, so that showing a shadow is not necessary for them. Due to the perspective, a menu with the same absolute size to the mid-air menu would look smaller on the screen. Therefore, we increased the size so that the rendered menu also takes up around 90 % of the device height when looking from the participant's seated position. The location of the menu stays the same throughout the session so that users can remember the location even if the table is not visible through the viewport when opening the menu. World-fixed menus did not perform as well as object-fixed menus in the study by Das and Borst [2010] and this design likely requires longer movement times since the menu is not automatically placed close to the object selected or at the fingertips of the user. Also, this is the only technique requiring a depth estimation to move to the intended location, potentially increasing the interaction times. However, the added haptic component of the surface might improve the interaction so much that users accept the longer movement and interaction times.

The menu techniques differ in respect to the interaction method.

While there are many other ways how menus could be implemented for the different interaction methods, the techniques we have chosen represent basic methods for each. Since the basic parameters for each menu are the same, this allows us to compare the effect of each interaction method on menu performance. This also becomes apparent when placing our menu techniques into the taxonomy by Dachselt and Hübner [2007], as the major differences are specific to the interaction method:

- *structural layout*: *one-handed touch* uses a list in an arc; all others use a radial pie menu;

- *reference* and *orientation*: 'virtual object' and 'user facing' for *two-handed touch*, *mid-air pen*, and *device* *pointer*; 'device' and 'on the side of the hand holding it' for *one-handed touch*; 'world' and 'table surface' for *surface*;

- *interaction device*: pen for *mid-air pen* and *surface*; pen+touchscreen for *one-handed touch*; touchscreen for *two-handed touch*; pen+device for *device pointer*

In the following section, we describe our study comparing the different menu techniques.

7.3 Comparing Menu Techniques

We compare the techniques in terms of success, time, movement, and subjective ratings.

To understand how the different interaction methods affect menu performance, we performed a study comparing the menu techniques in terms of successful selections, time for selection and movement of the device as well as subjective feedback regarding ease of use and comfort while using the menu. We included measurements on device movement since more movement could increase muscle strain making it harder to keep the device in hand. Particularly for *device pointer* we also wanted to see whether users preferred to use translation or rotation movement to select an item.

Participants had to select an emoji from a context menu.

The task for the participants was to select an emoji shown on the screen from a menu around a virtual cube. This represents the task in which a user opens a menu with the intended option in mind but not knowing where it will be located in the menu. Keeping everything aside from the interaction method for the item selection similar between each condition enables us to judge the impact of the different interaction methods. Our results can then be used as a starting point for an interaction designer to decide which menu technique to use.

7.3.1 Study Setup

We implemented the menu techniques starting from our open-source ARPen system (cf. Chapter 4). For our study,

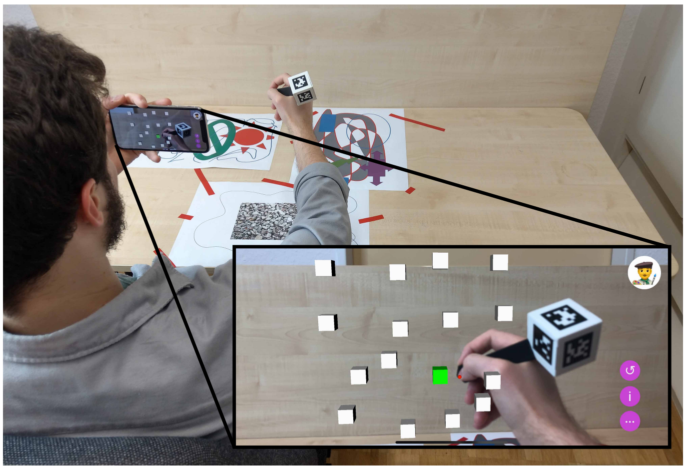


Figure 7.3: Participants had to open the menu for the green cube and select the emoji that was shown on the top right of the screen. The marker on the table ensured that the cube position was consistent between participants.

We implemented the menu techniques for the ARPen system.

16 cubes with an edge length of 3 cm were placed about 40 cm in front of the seated participant. The different cubes served as targets to open the menus for and we varied their position to avoid confounding effects from target placement. Since all techniques used ray-casting methods for the selection of targets, we did not vary the depth of target objects in the scene. A marker pattern on the table ensured that the position of the cubes stayed consistent between participants. For each trial, one of the cubes was highlighted in green, marking it as the target cube the menu should be opened for. Each cube was used twice as the target cube resulting in 32 trials per menu technique. The order of the target cubes was randomized.

Each menu contained a random selection of 8 emojis.

We used emojis as the items in the menus since they offer a strong visual and semantic differentiation suitable for our general task. The target emoji was shown in the top right corner of the screen. This emoji was randomly picked from one of eight categories: Faces, people, activities, flags, animals, food, travel, or objects. We chose the candidates for every emoji category so that they were easily distinguishable from each other. Each menu contained eight items based on the findings from related work [Kiger, 1984; Miller, 1981]. Random selections from the other categories of emojis were used to fill the remaining places in the menu. The placement of the emojis inside the menu was randomized but we ensured that the target emoji was placed at every position four times for each menu technique. Selecting an emoji closed the menu and showed the next target cube and target emoji regardless of whether the selection was correct or not. Opening the menu for a cube other than the current target cube showed a selection of emojis without the target emoji.

7.3.2 Study Procedure

Participants had to use the *pinkie*-grasp and select 32 emojis for each technique.

Participants sat in front of the table with the marker patterns. Each participant was asked to hold the phone in her non-dominant hand using her index finger and pinkie, leaving the thumb available for interaction with the touchscreen (*pinkie*-grasp from Chapter 4.3, see Figure 4.3, a). The participant was given time to test out the grip and the interaction before we explained the menu techniques and task of the study. Since our example usage scenario of mid-air modeling requires frequent specifications of mid-air points, we asked participants to keep the pen in their hand even for the *two-handed touch* technique which does not require the use of the pen otherwise. For each technique, we explained how the menu could be opened, described the visualization, and showed how an item can be selected. The participant could then try out the menu to familiarize herself with the interaction. Once the participant was confident in using the menu, she could select a button beside the marker to switch to the recording stage. The recording stage started with one additional test trial that was not recorded. This way, all recorded trials followed the same interaction sequence. If the participant needed a rest or stopped to comment on the interaction, we marked the current trial as an outlier and restarted the last trial. After each cube was used twice as the target for the menu, we asked the participant to rate how easy she thought the use of the menu was and how comfortable she felt during the interaction. Furthermore, we asked how she rated the combination of interaction techniques to open the menu and select an item as the interaction was not the same for every menu technique.

During the interaction with one menu technique, the moderator noted qualitative comments about the interaction. After having used all five menu techniques, we asked participants to rank them from best to worst. In total, each participant selected 160 emojis (32 trials$\times$5 *menu techniques*). The order of menu techniques was counterbalanced using a Latin square.

7.3.3 Measurements

We recorded the success of every trial by determining if the correct emoji was selected. Furthermore, we recorded the time for each trial separated into the time to select the target cube (*timeToMenu*) and the time from showing the menu to selecting the item (*timeToItem*). For the device movement we recorded the change of movement between frames for translation and rotation individually per axis with a frequency of 30Hz. The individual measurements per axis were summed up for the overall movement indicator.

The time measurement was split into *timeToMenu* and *timeToItem*.

For the subjective ratings for ease of use (*ease*), comfort of the interaction (*comfort*), and combination of selection techniques (*CoST*), we recorded 7-point Likert-Scale ratings for each technique as well as a *ranking* of all five techniques.

7.3.4 Evaluation

From our initial 15 participants, we excluded the data of one user as she was the only user who repeatedly selected the wrong target cube and then selected a wrong item. Also, she opened and closed menus more often than other users. We recruited another participant to fill the Latin square and have measurements for 15 participants (5 female, 22–54 years, M = 29 years, SD = 10.3 years, all right handed). Due to an error, two trials were not recorded. Overall, we collected measurements for 2398 item selections. For every participant, we summed up the number of successful selections for each menu technique and divided it by the number of repetitions to calculate the percentage

15 participants took part in the study.

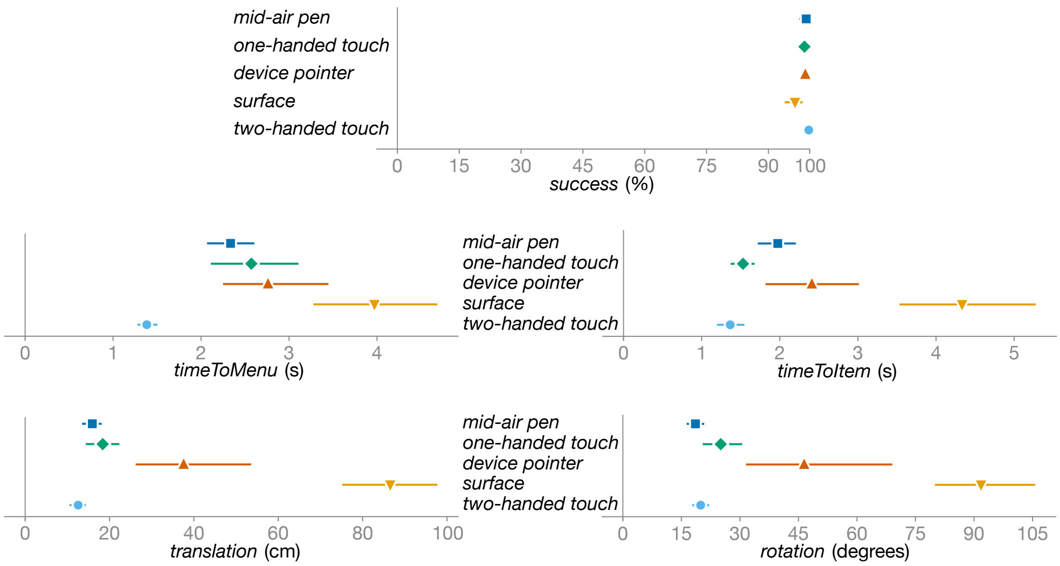


Figure 7.4: Results of the study comparing menu techniques. Means and 95 % CIs without within-subject adjustment.

We evaluate within-subject differences using confidence intervals.

of correct selections (*success*). We averaged the time to open the menu (*timeToMenu*), the time to select an item (*timeToItem*) as well as the translation and rotation movement (*translation*, *rotation*). We present our results based on the evaluation technique described in Appendix A. For each measurement of a menu technique, we subtracted the corresponding value of the baseline value (*two-handed touch*). This way, these results show the within-subject differences between the menu techniques and the baseline condition. For the following graphs, this means that values to the right of zero indicate that the baseline technique had less of the measured variable, while values to the left indicate higher values for the baseline compared with the other technique. For *success*, we did not subtract the percentages of the technique and the baseline but divided the success rate of the technique by the success rate of the baseline before calculating the differences. Therefore, we are able to report the differences not as percentage points but percentages. Since the interpretation of differences in Likert-scale data is not defined, we report the means and bootstrapped confidence intervals for the subjective ratings without computing the difference to the baseline.

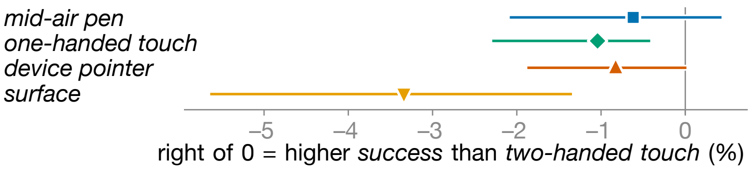
7.3.5 Results

In the following, we present the results of the study, first focusing on the measured data before reporting the subjective ratings of the participants and the qualitative remarks during the study. The overall confidence intervals without within-subject adjustment are shown in Figure 7.4.

Success, Time, and Device Movement

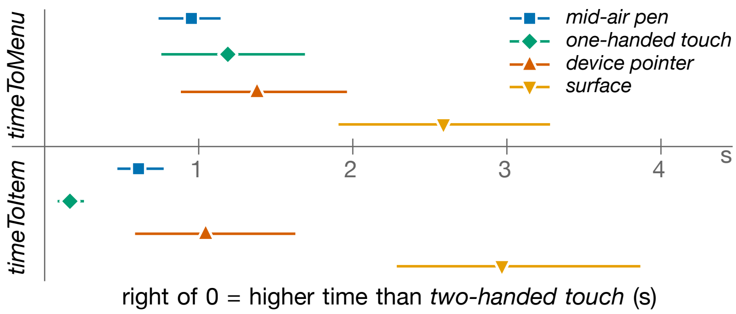
All techniques have a high success rate.

All techniques reach a high *success* rate (*two-handed touch* M = 99.79 %; CI [99.17 %, 100 %]) and there seem to be no differences between them with an indication that *surface* could be less successful.



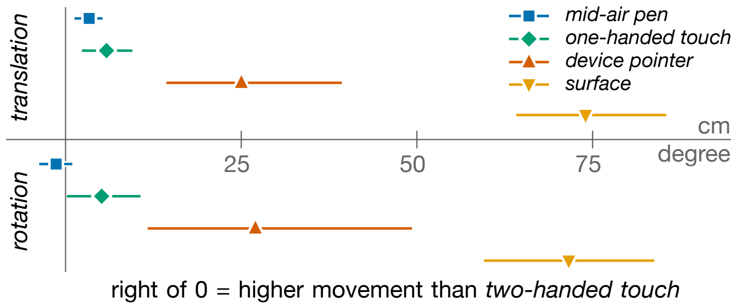
Using *two-handed touch* is faster and *surface* takes the longest.

For *timeToMenu*, the baseline condition (*two-handed touch*, M = 1.38 s, CI [1.3, 1.5]) shows faster results compared to the other techniques. Using the other techniques, participants had to use the pen-ray selection to open the menu. The difference of around 1 s to *mid-air pen* is similar to the difference found in our selection study (cf. Chapter 6.2). Even after the menu is opened, *surface* has the slowest *timeToItem* compared to the baseline (*two-handed touch*, M = 1.37 s, CI [1.2, 1.6]). *One-handed touch* seems to be close to the baseline with *mid-air pen* and *device pointer* following.



Surface & *device pointer* require more movement of the device.

Regarding the movement of the device after opening the menu, *translation* and *rotation* show that, as expected, *surface* shows the most movement of the techniques. *Device pointer* measurements show that, compared to the baseline, participants used both more *rotation* and *translation* movement. This indicates that not one type of movement was singularly used. Both *mid-air pen* and *one-handed touch* seem to be similar in the required movement of the device compared with the baseline with a trend to more translation. On average, participants translated the device by 12.6 cm (CI [10.6, 14.6]) and rotated the device by 19.9 degrees (CI [17.8, 22.6]) in the baseline condition.



Ease, Comfort, Combination of Selection Techniques, and Ranking

Two-handed touch achieves the highest subjective ratings.

Looking at the subjective ratings for *ease* of use of the techniques (cf. Figure 7.5, top), we see that the *two-handed touch* baseline was rated as the easiest technique to use. The other techniques show ratings similar to each other but the tendency goes towards the order *mid-air pen*, *one-handed touch*, *device pointer*, and *surface*. Ratings regarding the *comfort* are even closer together than the ratings for *ease* of use (cf. Figure 7.5, middle). Especially *mid-air pen*, *one-handed touch*, and *device pointer* move closer together and also *two-handed touch* was rated more similar to them while still being rated more comfortable. The results of asking participants to rate the combination of selection techniques (*CoST*) to open the menu and select the item shows that the techniques that use the same method for both (*two-handed touch*, *mid-air pen*) achieve the highest ratings (cf. Figure 7.5, bottom) followed

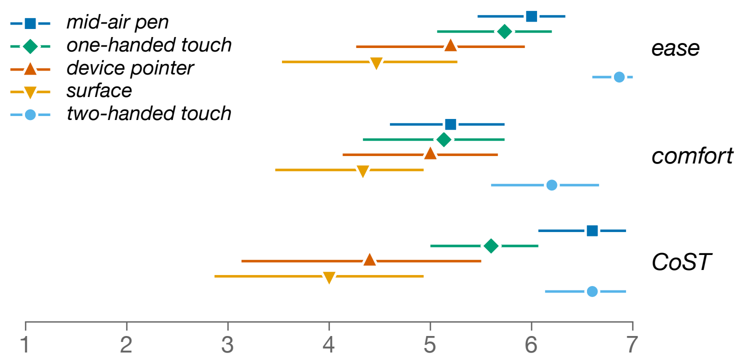


Figure 7.5: Subjective ratings for *ease* (top), *comfort* (middle), and combination of selection techniques (*CoST*, bottom). *Two-handed touch* has the highest ratings while *surface* seems to be less easy and comfortable to use.

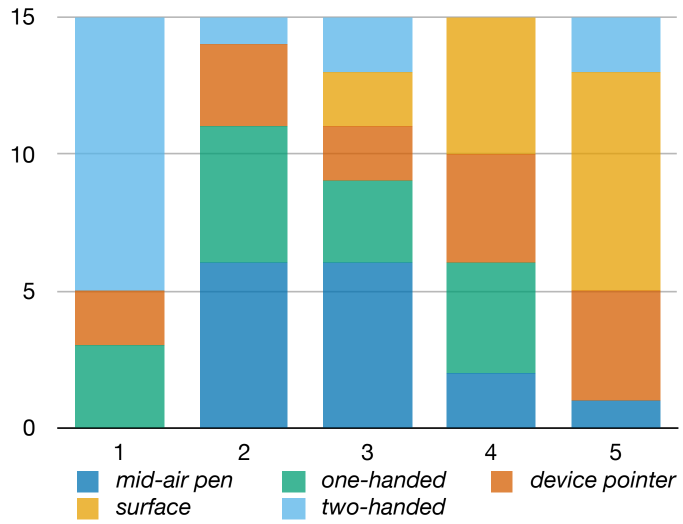


Figure 7.6: Subjective ranking of the menu techniques. *Two-handed touch* seems to be most preferred while *surface* is often placed in the last position. *One-handed touch* and *mid-air pen* are placed high and *device pointer* at all positions.

by *one-handed touch*. *Device pointer* and *surface* received lower ratings. The *ranking* of techniques shows that participants seem to prefer the baseline condition (*two-handed touch*) while placing *surface* often in the last place (cf. Figure 7.6). Both *one-handed touch* and *mid-air pen* are placed more towards the top of the ranking while *device pointer* was placed at every place of the ranking.

Participants prefer *two-handed touch* before *one-handed touch* & *mid-air pen*.

Qualitative Remarks

The world-stabilized menu is not liked by participants.

While *surface* seemed to be less liked by participants, several mentioned that they liked the physical guidance of the surface, but only two rested their hand on the surface while approaching the target item. However, participants commented that they did not like the switch in context between mid-air target and menu on the surface. For *device pointer*, seven participants explained confusion about which device/cursor to use for which step even though only one cursor was visible at any time. On the other hand, other participants assumed that the performance of this technique could improve a lot when using it for a longer time, with three participants explicitly stating that the learning process was very fast. The biggest commented issue for *one-handed touch* was that participants had trouble holding the phone in the grip and simultaneously using the thumb to interact. Others, however, mentioned that they liked that the menu is directly under the thumb and that the shape would help to select the menu items. For *one-handed touch*, *two-handed touch*, and *mid-air pen*, participants said that they found the techniques easy to use and especially *two-handed touch* would be what they are used to. However, two participants mentioned that it would be annoying to switch between mid-air interaction with the pen and touchscreen interaction frequently. Similar to our previous studies, participants noted that they felt fatigue in their arms since they had to hold both phone and pen in mid-air for most parts of the interaction. This shows that this interaction is not suitable for prolonged use but more for short periods of time.

Frequent switches between mid-air and touchscreen could be annoying.

7.4 Discussion

Two-handed touch is the fastest and most preferred technique.

Our results indicate that the baseline condition of *two-handed touch* is the fastest and most preferred technique to use to interact with menus in 3D environments on a handheld device. This is interesting in comparison to our previous selection study (Chapter 6.2) in which the pen-ray interaction was more similar to the two-handed interaction. On the other hand, the one-handed interaction was not rated high in the selection study but achieved good results and ratings in this study. This means that fixing the menu to the side of the screen improves this technique. *Mid-air pen* and *one-handed touch* are not far behind *two-handed touch* as they showed not only similar success rates but did not differ much in terms of the device movement and also achieved sufficiently close selection times. Only for the time to open a menu, *two-handed touch* seems to be about one second faster which is in line with the findings from our selection study. In a use case where a lot of the interaction is happening mid-air, such as sketching or manipulating objects, frequent switches between mid-air and touchscreen interaction will limit this time benefit of *two-handed touch*. In such cases, staying within the context in mid-air could prove to be advantageous. *One-handed touch* and *mid-air pen* performed largely comparable. The differences in the time to open the menu between the techniques that use the pen-ray selection to trigger the menu can be explained by the different behavior after an item has been selected and the new target cube is highlighted: For both *one-handed touch* and *device pointer*, the focus of the user is not on the mid-air pen, requiring the user to find the pen tip again. For *surface*, participants had to lift the phone from the surface to see the mid-air cubes again. While users preferred using the same method for opening the menu and selecting an item in *mid-air pen*, *one-handed touch* seems to be a bit faster for selecting the menu item and it can be used independently of the pen position in mid-air. This could be advantageous for more complex menu input so that users can rest their hands and are not required to keep them lifted during the whole interaction. This point could also be a benefit for *surface* which did not perform as well as other techniques in our metrics. We expected that this technique requires more movement and time for the interaction due to the world-stabilized style of the menu and also since it is the only menu technique that requires a depth estimation by the user to move to the intended location. On the other hand, the physical surface provides haptic feedback during the interaction and we wanted to see whether this could compete with the movement and time increases. While participants mentioned that they liked the physical guidance of the surface, they gave this condition lower ratings and a lower ranking. This menu technique might be helpful to

Fixing a menu to the screen or interacting with the mid-air pen achieves good results.

Only *surface* requires a depth estimation. The physical surface does not cancel out the drawbacks.

use in scenarios where opening and closing the menu does not happen frequently and when the menu interaction is followed by an interaction for which the physical surface provides an advantage. An example could be the situation where a hand drawing should be projected on a plane or object in the scene. Selecting the object mid-air could then lead to the menu specifying the drawing properties using the surface menu before using the physical surface to draw. *Device pointer* achieved mixed results indicated also in the spread across all placements in the ranking. While success rate and interaction times are comparable or close behind the other techniques, using this techniques requires more movement of the device and participants mentioned that it is more difficult to understand this technique. On the other hand, participants also mentioned that the interaction would be fast to learn. Since it does not require moving the pen or the focus of the user, it could be an interesting technique for routine and quick selections that can then be selected blindly by 'flicking' the device. An example could be to choose that the selected object should be moved. After selecting this option with *device pointer*, the object snaps to the pen tip so that it can be placed.

We present potential use cases for each menu technique.

For menus, there is no "one size fits all", and designers will have to weigh the different options for their specific scenario. Our results give them a starting point, e.g., to decide whether an increase in item selection time of about one second is acceptable if the device pointer otherwise fits their interaction scenario. The measurements do not directly disqualify any menu technique and for each there could be a scenario where it is the most sensible choice. Following, we present general suggestions as well as potential use cases, based also on the qualitative feedback from our participants, in which choosing a menu technique other than the "fastest" or "most accurate" could make sense:

- If the interaction happens on the touchscreen and only occasionally mid-air, use the standard *two-handed touch*.
- If most of the interaction happens mid-air, consider using *one-handed touch* and *mid-air pen*. The preference for either depends on the use case.

- For quick and routine actions, consider *device pointer*. Users will be able to keep the mid-air pen and their focus at their current position.
- If the interaction after using the menu benefits from a physical surface, using *surface* provides the benefits from the haptic surface, which could outweigh its drawbacks such as longer interaction times.

Our study provides first insights into the effect of different interaction methods on menu performance in Handheld AR systems. To encourage further research and replication, we provide both our software and data.[1]

7.5 Summary & Future Work

We compared different menu techniques. *Two-handed touch* achieves the best results, followed by *mid-air pen* & *one-handed*.

In this chapter, we presented the results of a study comparing menu techniques for Handheld AR applications using a mid-air pointing device. The menu techniques were sampled to account for the different interaction methods such a system offers: mid-air, touchscreen, physical environment, and movement of the device itself. We found that the standard technique of *two-handed touch* seems to achieve the best results overall. However, if most of the interaction happens in mid-air, switching to the touchscreen can become annoying. In such cases, a menu in mid-air (*mid-air pen*) or a menu on the side of the touchscreen to be operated by the hand holding the smartphone (*one-handed touch*) seems to be the preferable options. Our results provide interaction designers with an estimation of the differences between the menu techniques so that they can pick the most suitable for their application.

Future work could study menu techniques in different scenarios.

We have used a simple scenario for our study to evaluate the general impact of the different interaction methods on menu performance and reduce the impact of extraneous factors as much as possible. Future studies could increase the external validity by studying the menu techniques in scenarios closer to the real world. This could mean different

[1] hci.rwth-aachen.de/armenus (accessed 28.05.2021)

surroundings, such as cars and kitchens, as well as different tasks, such as menu interactions embedded in a modeling or drawing task. While we focused on menus that use the main interaction capabilities of a Handheld AR system with a mid-air pen, there are numerous ways how such a system could be extended. These extensions, for example back-of-device or voice interaction, could also be used for menu interactions and we encourage further exploration in this direction. Other studies in this area could adjust different aspects specified by the taxonomy from Dachselt and Hübner [2007], and study their impact. Our study investigated context menus, but global menus are important as well and their combination is required for an application. For example, the ARPen system uses *one-handed touch* for the selection of different plugins and the results from this study suggest that this is a suitable technique. Displaying a context menu in a similar way would be possible, but using *mid-air pen* could clearer separate a global from a context menu. Further studies could investigate the combination possibilities. Another interesting direction, for example, is to adjust the hierarchical nature to require multiple menu levels and the navigation between them—a possible visualization for additional menu levels is discussed by Gebhardt et al. [2013]. Other promising research questions include matching different menu techniques to specific tasks such as browsing through 3D objects, and then combining these different menus in a coherent and usable way.

Moving on from menus as an example of a *system control* task, we address the cognitive component of *navigation*, called *wayfinding*, in the next chapter. We design and evaluate different depth visualization techniques that help in determining the position of the mid-air pen relative to other objects in the scene.

Chapter 8

Depth Visualizations

Understanding the spatial arrangement of objects is crucial for *situated modeling*.

In the previous chapters, we have designed and evaluated interaction techniques for the areas of *selection & manipulation* and *system control*. In this chapter, we address the cognitive component of *navigation*, which LaViola et al. [2017] call *wayfinding*. *Wayfinding* includes understanding the arrangements of objects and the user's own position in the environment, for example, in order to move to a new position. However, perceiving the depth information of a virtual object in AR and VR is problematic (cf. Chapter 2.3). This creates issues for *situated modeling* tasks in Handheld AR, because even though it is relatively simple to find a specific location in the real world and move the pen to it, it is considerably harder to find that spot if the object to target is virtual. For selection tasks, this leads to ray-casting methods being preferred, which select the first object that is hit by a ray shot from a touchpoint on the device's screen or through the tip of the mid-air pen (cf. Chapter 6.2).

At the same time, actually finding and moving the pen to a specific location in the environment is a central task for

Publications: The work presented in this chapter has been done in collaboration with Adrian Wagner, Simon Voelker, and Jan Borchers. It is published as a full paper at ACM CHI '20 [Wacker et al., 2020b]. The author of this dissertation developed the research idea and is the main author of the paper. He designed the techniques as well as planned and executed the study together with Adrian Wagner. The author also analyzed all data from the study. Most sections in this chapter are taken from the full paper publication.

Aligning the pen with a virtual object is required for *isomorphic* drawing.

mid-air modeling systems. For example, to sketch a line from one virtual object to another, a user first has to find the correct spot on the virtual object from where she wants to draw the line. While ray-casting techniques could place a remote cursor at the intersection and map pen movement onto that cursor to create a *non-isomorphic* drawing technique, this approach takes away a key advantage of AR modeling, because now the position of the real world pen does not align with the position of the virtual pen anymore, and haptic properties of the physical environment can no longer be used for guidance.

We designed and evaluated different depth visualizations.

To tackle this fundamental problem of mid-air interaction in Handheld AR, we defined and implemented different visualization techniques that show the position of the pen in relation to surrounding objects, and compared them to a baseline condition with no such visualization. We ran our comparisons in scenes with both solid and wireframe objects. Based on our results, we provide design considerations for visualizing the 3D position of a mid-air pointing device in Handheld AR.

Our contributions in this chapter therefore are:

- the definition and comparison of different visualization techniques to show the 3D position of a mid-air pointing device in relation to surrounding objects in Handheld AR,

- design implications accounting for the advantages and disadvantages of these visualizations.

Following this introduction, we give an overview of related work in depth perception in virtual environments. We then present the different visualization styles we designed, and how we implemented them, followed by our study setup and procedure. After summarizing the results of our study and discussing their design implications, we conclude with an outlook on future work.

8.1 Related Work

In order to create a realistic representation of the 3D position of a virtual object in virtual environments, the system needs to simulate depth cues (cf. Chapter 2.3). Most studies on depth cues focus on stereoscopic systems [e.g., Ellis and Menges, 1998; Jurgens et al., 2006; Livingston et al., 2009]. Those dealing with Handheld AR systems tend to look at farther distances, for applications like X-ray viewers or AR browsers [e.g., Dey and Sandor, 2014; Swan et al., 2017; Wither and Höllerer, 2005]. Wither and Höllerer [2005] compared different depth cues in a headworn AR environment for distances of more than 20 meters. They compared different visualizations in which the size of a cursor varied with the distance, shadow planes showed the layout of objects, a top-down overview of the scene was provided, or the color of objects in the scene changed based on the current cursor position. They found that other objects in the scene improve depth accuracy, but the visualizations themselves did not improve accuracy as much as expected—something they attribute in part to the visualization system used. The top-down view was preferred by users, and performed well, especially for the nearer objects in their study. The color visualization also received positive user responses. Other studies showed that one of the most effective cues to indicate the depth of a virtual object is displaying its shadow [Diaz et al., 2017; Drascic and Milgram, 1996; Hubona et al., 1999; Wanger et al., 1992].

Most studies on depth cues focus on stereoscopic systems.

Studies of depth perception in virtual 3D environments differentiate between *egocentric* and *exocentric* depth perception [e.g., Dey et al., 2012; Dey and Sandor, 2014; Swan et al., 2006]. Egocentric depth perception measures the distance of an object from the viewer, while exocentric depth perception looks at the distance between different objects in the scene. Most research so far has focused on egocentric depth perception [Dey et al., 2012; Jones et al., 2008].

Studies differentiate between *egocentric* & *exocentric* depth perception.

In our study, we look at a task for which both egocentric and exocentric depth perception are important. To reach a specific point on a virtual object with her mid-air pen, a user first has to evaluate the depth of the object relative to

We study a combo of egocentric and exocentric depth perception.

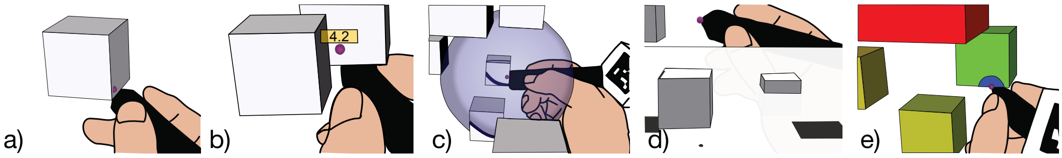


Figure 8.1: The visualization techniques used in our study. a) *minVis* as the baseline, b) *depth ray*, c) *bubble*, d) *shadow*, e) *heatmap*.

her own position, and then continuously evaluate the position of the mid-air pen tip relative to the target position to reach the intended location in 3D space. Our study is the first project to evaluate different depth visualization techniques for such a task using a Handheld AR system with a mid-air pointing device.

8.2 Visualization Styles

We designed 5 *depth visualizations*.

We define five visualization styles to indicate the 3D position of the pen tip relative to other objects in its environment. Our baseline is a minimal visualization without any extra depth information, while four depth visualizations build on this baseline to provide more information for the depth dimension (cf. Figure 8.1).

minVis: the sphere on the pen tip interacts with the environment.

Minimal Visualization: In our selection study (Chapter 6.2), we found that displaying no indication of the pen tip does not allow participants to select objects in a 3D scene: only 3 % of targets were successfully selected. This indicated that it would not enable users to select a point on a virtual object either. We also found that rendering a small sphere at the tip of the tracked pen improved selection performance because it represented another object in the scene. Like every other virtual object, this sphere was not rendered if inside or behind another virtual object. Other than that, it did not provide additional information about the position of the pen. For the depth visualizations, we define this as the minimum viable visualization to estimate the position of the pen in the scene.

We use the depth cue of occlusion only for the tip of the pen, not for the whole pen and hand holding it. One reason for this is that current smartphones cannot reliably track moving physical objects in the scene to calculate how they occlude virtual objects (cf. Chapter 2.3). While this technical limitation could be overcome with a lab study, modeling and rendering the hand would also occlude a large part of the virtual information on the smartphone's small screen, and the effects of this on scene perception and the overall user experience will need to be studied first. Rendering only the pen without the hand into the scene would be easier to achieve technically, but the perception of ownership is affected by the style of visualization [Rosa et al., 2019]. Since the pen would be rendered in front of the hand holding it, "cutting off" its thumb, it would create an unnatural visualization that would likely affect the sense of *holding* the device [Schwind et al., 2017]. Until perceptual issues such as these are answered and hand-tracking capabilities increase, we decided to focus on the visualization of the position of the pen tip for this work. We present potential future research directions exploring these issues at the end of this paper. Not relying on a visualization of pen or hand, however, also makes our results easier to apply to other input devices that support the fundamental interaction of specifying a mid-air position in Handheld AR.

We use occlusion only for the pen tip, not for the pen and the hand holding it.

All following techniques build up from this minimum viable visualization. For each visualization, we provided users with a way to toggle it on and off using a button on the pen. This way, the additional visualization could be brought up only when needed, without cluttering the scene otherwise. This is important to keep in mind when interpreting our findings.

Users can toggle the state of the following depth visualizations between on & off.

Depth Ray: Since ray-casting methods have shown good results for selecting objects, we created a method based on the ray from the camera through the tip of the pen into the scene. We display the distance of the pen tip to the next object behind it in cm. This lets users judge the distance of the pen to the objects behind it, which, for the problematic task of moving the pen to the correct 3D position, can help to decide whether large or fine movements in the depth direc-

Depth ray: the distance to the object behind the pen tip is shown in a label.

tion are needed to reach the intended position. However, this visualization does not give any additional information in other directions than behind the pen tip.

Bubble: a bubble grows from the tip and intersections with objects are visible.

Bubble: To show distance information in all directions, we defined a bubble visualization. When triggered, a semi-transparent bubble linearly grows outward from the current pen tip position over one second, while remaining centered around the pen tip. During the increase in size, the intersections of the bubble shell with objects in the scene are highlighted. This means that objects farther away from the center of the bubble are intersected later than closer objects and the intersections indicate a larger size of the bubble. Other than that, the bubble behaves like other virtual object in the scene, and is therefore occluded by objects in front of its shell and occludes objects behind its shell.

Shadow: objects and the pen tip cast shadows on a plane below the viewport.

Shadow: Previous studies have shown that shadows improve depth perception in virtual environments [e.g., Drascic and Milgram, 1996; Diaz et al., 2017; Wanger et al., 1992]. However, the viewing area in Handheld AR is very limited, and the next surface to project shadows on, such as the table surface, is often outside the camera image, so seeing the shadow would require moving the smartphone camera. Therefore, we chose to display the shadows on an artificial horizontal plane rendered slightly lower than the current viewing height. The light source is placed above the scene, so that all items cast their shadow downwards onto each other or the artificial plane. This includes the small sphere at the pen tip.

Heatmap: objects are colored based on their distance to the pen tip.

Heatmap: To show the mid-air position of the pen relative to all objects in the scene, a *heatmap* visualization can be used. Similarly to the "Marker Color" technique by Wither and Höllerer [2005], it shades all objects by replacing their original color with a color indicating their distance to the pen tip. For our naive implementation we linearly transitioned from full green to full red in RGB space over 200 mm. Far away objects are colored in red, and the shading gradually changes to green the closer the object is to the pen tip. In addition, we colored surface areas within 10 mm of the pen tip in blue to indicate the closest surfaces. We checked the visualization with a red-green color vision impaired user to confirm that the difference between the red and green color was strong enough. A linear gradient between colors in RGB space is not perceived as linear by humans [Lissner and Urban, 2012] but it can provide us with a general impression of the effectiveness of such a visualization by showing if users are able to apply the mapping that red is farther away from the target than green.

Aside from the five depth visualization techniques described above, we designed and implemented several other visualizations, such as displaying grid lines, adding light sources to the pen tip or camera to light the scene, and rendering lines between the pen tip and a number of closest objects. However, pilot studies showed that these techniques were harder to understand and interpret than the visualization techniques we selected for this study.

We designed other visualizations but did not include them in our study.

8.3 Study of Visualization Styles

In order to test the performance of our proposed visualization styles, we compared them to the baseline condition that uses minimal visualization (*minVis*). Many 3D pointing operations require identifying the distance of the input device to its surroundings, e.g., to arrange objects or to create a connection between them. We tested the case of finding the correct location on a virtual object in mid-air. This task setup is basic enough to be applicable to many other settings since the perception issues on mobile devices stay the same. All of our techniques could be applied to a physical environment and physical objects as well if all the elements in the scene are tracked. The performance of each technique would likely increase then, because physical effects, such as peripheral vision around the device or haptic feedback from objects, could be used to support moving to the intended location. As mid-air modeling systems can feature both solid objects and wireframe models (or sketched lines), we included both cases, *solid* and *wireframe*.

We tested the case of finding the correct location on a virtual object in mid-air.

We included solid and wireframe objects.

8.3.1 Study Setup

We implemented the visualizations for the ARPen. Users can toggle the visualization by pressing a button.

We implemented our visualizations for our open-source ARPen system (cf. Chapter 4). Users can toggle the current visualization on or off by pressing a button on the pen—our pilot users preferred this to having to keep holding the button down to see the visualization. Our implementation only took surrounding virtual objects into account but the visualizations could be applied to physical objects as well. To enable the visualizations for physical objects, the position and geometry of these objects needs to be tracked.

Participants had to move the pen to a position indicated on a target object and start drawing.

The task for each trial in our study was to move the pen to a specific location on a virtual cube object and start drawing a line from that location. The interaction volume for our study was $400 \times 400 \times 400$ mm in front of the participant, so that the whole area was in arm's reach. To keep the position of the interaction volume constant between participants, we used a marker on the table to fix its position. We separated this volume into $3 \times 3 \times 3 = 27$ areas, and placed a virtual cube into each area, with an edge length of either 30 or 40 mm to prevent the depth cue of *relative size* to influence our task. The cube's position in its area was randomized for each new condition, while ensuring a gap of at least 20 mm between two adjacent cubes. For each trial, we showed eight cubes and marked a location on a target cube to indicate where to move the pen to. Since this location should be visible to the user, we first excluded cube sides that were invisible when looking from a centered position in front of the interaction volume—this included back-facing sides for all cubes and, e.g., the right-facing side of a cube on the right side of the interaction volume. On a random remaining side, we chose a random position as the target position for this trial. In the case of *wireframe* objects, this position was limited to the edges of the surface. We indicated this position with a purple sphere clearly visible against the rest of the cube. Participants had to reach this sphere with the pen as precisely as possible, and start drawing a line from this position. This line was a placeholder for an action in a real modeling task; so its shape was not relevant for our evaluation and we did not analyze it.

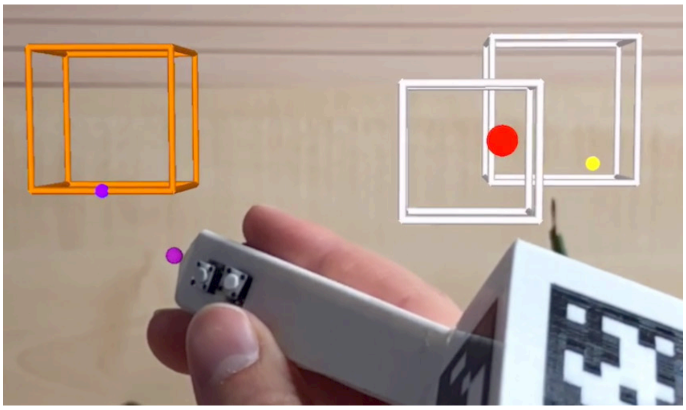


Figure 8.2: *Wireframe* objects in the study. Before starting the trial, participants had to move the smartphone to align the yellow and red spheres. The target location on the cube was indicated by a purple sphere.

To control the starting position for each trial, we added two spheres to the scene, one at the center of the near side, one at the center of the far side of the interaction volume (cf. Figure 8.2). Participants had to align these spheres on the screen and lift the pen into the camera view to start each new trial in a comparable situation.

The starting position for each trial was controlled by aligning two spheres.

8.3.2 Study Procedure

Participants sat in front of a table with the marker. They were asked to hold the device in their non-dominant hand in the *pinkie*-grip: resting on the pinkie and gripped by the index finger (cf. Figure 4.3, a). Participants could familiarize themselves with the grip and drawing with the mid-air pen before we introduced the task and the visualization techniques. Before each new technique, we explained how it works, and let them familiarize themselves with it for up to three minutes. Once they were confident to continue, we started recording the trials. We chose the cube with the target marking randomly from the 27 cubes, and repeated the trial 16 times, making sure that no cube was selected twice as the target cube. After completing the 16 repetitions for a condition, we asked participants to rate their confidence in the ability to find the intended location, the helpfulness

Participants used each condition 16 times.

of the visualization, and how easy it was to comprehend. While they were completing the trials for a visualization technique, the moderator captured qualitative comments and observations about the interaction. After using all five different visualization techniques, we asked participants to rank them by preference from best to worst. Each participant selected 160 positions in mid-air (5 *depth visualizations* (*minVis*, *depth ray*, *bubble*, *shadow*, *heatmap*) $\times$ 2 object styles (*solid*, *wireframe*) $\times$ 16 repetitions). The order of visualization techniques and object styles was counterbalanced using a Latin square.

8.3.3 Measurements

We measured deviation, time, duration the visualization was active, and subjective ratings.

For each trial, we recorded how far the initial drawing position deviated from the intended position indicated by the marking (*distanceToTarget*). We also measured the time from the beginning of the trial to the beginning of the first drawing operation (*timeToTarget*). To evaluate how often the visualizations were used to find the intended position, we recorded how long the visualization was active, and from this calculated a relative duration (*visualizationPercentage*). The times before aligning the spheres or after completing a trial were not recorded to give the participants time for comments. If participants needed a rest or stopped during a movement task, e.g., to provide a longer comment, we repeated the last trial. This happened 49 times in our study. For *confidence*, *helpfulness*, and *comprehensibility*, participants rated the techniques on a 7-point Likert-scale, and ranked them from best to worst.

8.3.4 Evaluation

We recruited 10 participants (4 female, 21–28 years, M = 25 years, SD = 1.8 years, all right-handed, no self-reported color-vision deficiencies). Overall, we recorded information for 1599 finding operations, as one trial was not recorded due to an issue with the device. For every participant, visualization, and object style, we averaged the

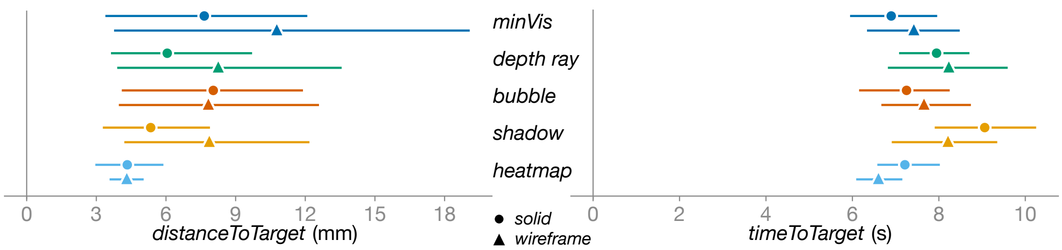


Figure 8.3: Results for *distanceToTarget* and *timeToTarget*. Means and 95 % CIs without within-subject adjustment.

deviations (*distanceToTarget*), time to move to the intended location (*timeToTarget*), and percentage of time the visualization was active (*visualizationPercentage*). We calculated confidence intervals and effect sizes based on the procedure described in Appendix A. These effect sizes show the within-subject differences between the measurements of the visualization techniques and the baseline condition (*minVis*). Consequently, in the following graphs a value to the right of '0' indicates more of the measured variable for the visualization technique compared to the baseline, while a value to the left indicates less. As the subjective ratings were recorded as Likert-scale ratings for which the meaning of differences is not defined, we did not compute the difference to the baseline but report the means and bootstrapped confidence intervals. Individual results for the visualization techniques, such as measurements for the overall results, are calculated and reported the same way.

We evaluate within-subject differences using confidence intervals.

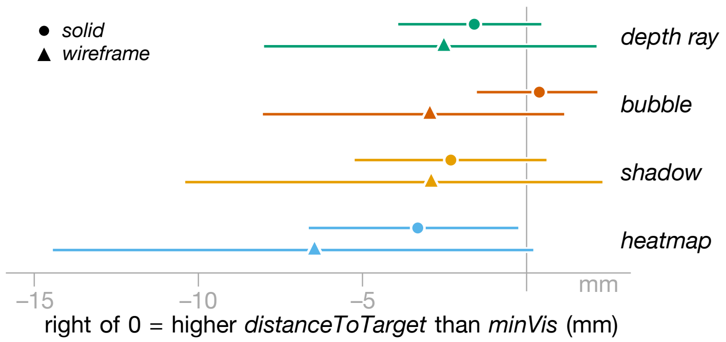
8.3.5 Results

We report the results of our study starting with recorded measurements before continuing with subjective ratings. The overall confidence intervals without within-subject adjustment are shown in Figure 8.3. Following, we analyze within-subject differences for our measurements.

Distance to Target

No clear advantage of the visualizations is apparent.

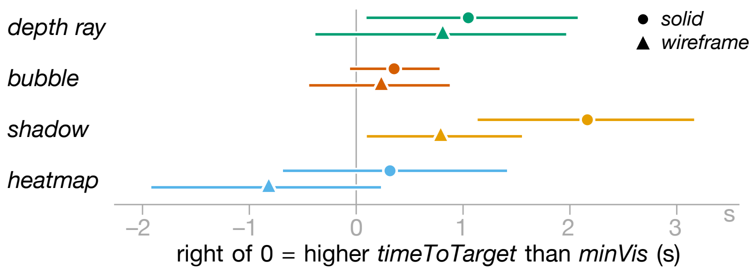
Regarding the distance of the first drawing operation to the target location, no clear differences of the visualizations compared to the baseline are apparent. That most of the means and the confidence intervals are to the left of 0, especially for *heatmap*, indicates smaller distances to the target compared to the baseline of *minVis* (M = 9.21 mm; CI [3.87, 15.86]). The larger spread of confidence intervals in the *wireframe* condition is linked to a larger spread particularly in the *minVis* condition. The individual distance results, especially for *heatmap*, have narrower confidence intervals (cf. Figure 8.3). The style of objects, *solid* or *wireframe*, shows the greatest difference relative to *minVis* in the *bubble* condition.



Time to Target

Using a visualization seems to mostly increase *timeToTarget*.

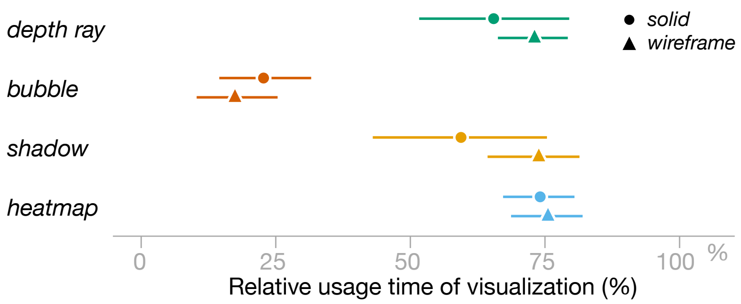
Evaluating *timeToTarget* shows that using the visualization techniques, except for *heatmap*, seems to increase the time required to reach the intended target compared to *minVis* (M = 7.16 s; CI [6.16, 8.18]). Especially for *shadow+solid*, users seem to take more time compared to the baseline. *Heatmap* performs similar to *minVis* and is perhaps marginally faster for *wireframe* objects.



Percentage that the Visualization was Active

Since there was no additional assistance in the *minVis* condition, the graph shows the individual percentages that each visualization was active while approaching the target. The diagram clearly shows that *bubble* was used considerably less than the other visualizations. All other visualizations were used a comparable percentage of the time. For *depth ray* and *shadow*, it seems as if the visualization is used less for *solid* objects. In 133 trials, participants did not activate the visualization technique at all. Most of these trials belong to the *bubble* condition (87 trials) followed by *shadow* (33 trials), *depth ray* (10 trials), and *heatmap* (3 trials).

The *bubble* visualization is used the least.



Confidence, Helpfulness, Comprehensibility

Participants' subjective ratings show that they felt the most *confidence* to be able to find the location when using *heatmap* (cf. Figure 8.4, top). *Shadow* and *depth ray* seem to be rated lower, with *bubble* and *minVis* even more so. The perceived *helpfulness* of the visualization follows a similar pattern (cf. Figure 8.4, middle). All participants gave *heatmap* the top

Participants rate and rank *heatmap* the highest, followed by *shadow* & *depth ray*.

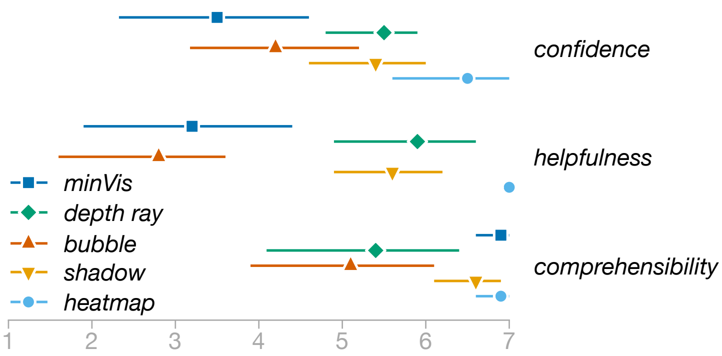


Figure 8.4: Subjective ratings for confidence, helpfulness, and comprehensibility. *Heatmap* achieved the high ratings overall, followed by *shadow* and *depth ray*. For comprehensibility, *minVis* is also high, with *depth ray* being rated lower.

mark, while there is no difference in ratings of *depth ray* and *shadow*. *Bubble* and *minVis* clearly scored the lowest. *Comprehensibility* of *minVis*, *heatmap*, and *shadow* was rated very high (cf. Figure 8.4, bottom). *Depth ray* and *bubble* were rated lower. Most of our participants ranked *heatmap* as their preferred visualization technique (cf. Figure 8.5), followed by *shadow* and *depth ray* with similar rankings. *Bubble* and *minVis* are mostly ranked on the last places.

Qualitative Comments

minVis already provides good information for the final alignment.

Participants noted that the visualization of the pen tip and its interaction with the environment provided good information during the final approach to the target object, as the tip would disappear once it was inside the object (P3: *"The pen tip entering the cube made it feel like I could precisely estimate the position but [it] took long"*). Comments regarding *heatmap* were generally positive, and participants noted that they would also get information if the pen tip was inside a *solid* object. For *depth ray*, *wireframe* created problems: Participants mentioned that it was hard to differentiate between lines that were rendered close to each other but varied in distance (P6: *"The full cubes helped a lot more, because you exactly know what surface the measurement is based on"*). Participants also stated that it was difficult to inter-

Interpreting *depth ray* is difficult for *wireframe* objects.

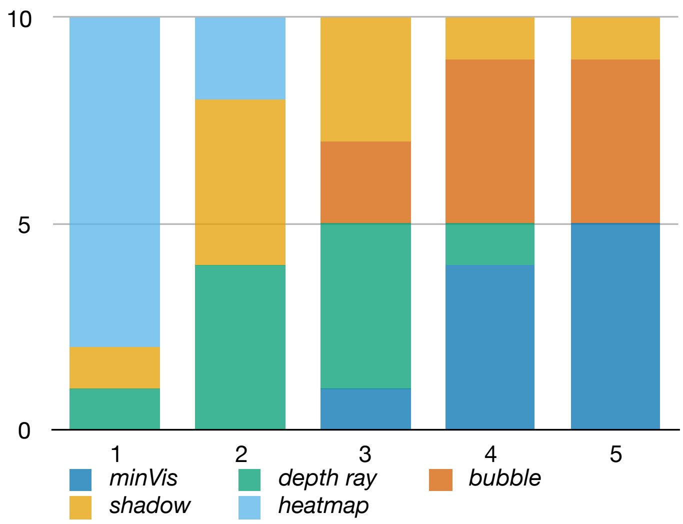


Figure 8.5: Subjective ranking of the visualization techniques. *Heatmap* is ranked first most frequently, with *depth ray* and *shadow* contending for rank two and three. *Bubble* and *minVis* share the last two places.

pret the numerical value, and that therefore it was often used just as a trend of movement. In the *shadow* visualization, participants mentioned that they preferred to use the technique for *wireframe* objects, as it also provided information *inside* the object, whereas shadows of *solid* objects often overlapped each other (P5: *"Filled cubes made finding the shadow of the pen tip more difficult"*). Participants also said that tilting the device downward to see the shadow plane limited their viewing options with the device. *Bubble* received mostly critical comments, stating that the visualization was hard to understand (P5: *"The bubble is not helpful at all, I do not get it"*).

Shadows of *solid* cubes overlap each other.

8.4 Discussion & Design Implications

The results of our study indicate that *bubble* is not a good visualization technique to show the position of a mid-air pen, as participants rated the helpfulness of the visualization low and mentioned that they were not able to comprehend the depth cue. This also becomes apparent in the

Bubble is hard to understand.

Heatmap achieves the best results overall.

high number of trials in which participants did not even activate the visualization. *Heatmap* achieved the highest ratings and had good performance results, as it showed the greatest possibility for improving accuracy and movement time. While *depth ray* only showed the distance in one specific direction, and *shadow* required the user to tilt their device to see the visualization, *heatmap* and *bubble* showed the position of the pen tip in all directions. However, *bubble* did this in a non-permanent, time-based way, whereas *heatmap* was visible all the time and on all objects in the scene while active. Coloring the whole scene worked well in our setup. If, however, the texture of an object provides the target that the user wants to move the pen to, then this might require the user to switch off the visualization for the final approach. If such a case occurs regularly, the application designer might want to switch to other visualization techniques such as *shadow* or *depth ray*. We suggest additional solutions in Chapter 8.5.

Depth ray & *shadow* perform similar.

Depth ray and *shadow* performed quite similar, and also received comparable ratings. While *depth ray* seems to be slightly faster than *shadow*, *shadow* scored higher in comprehensibility. However, the qualitative comments show that the usefulness of these techniques was seen differently based on the style of objects in the scene. While *shadow* was considered more useful in scenes with *wireframe* objects, because the shadows do not overlap or occlude the shadow of the pen tip, *depth ray* was said to work better in scenes with *solid* objects. A reason for this is that it is more difficult to understand which element in the scene the current distance is referring to in scenes with *wireframe* objects. The additional time required for *shadow* and *depth ray* could be due to the need to interpret the visualization: the numeric value for *depth ray*, and the shadow on the plane beneath the smartphone for *shadow*. *Heatmap* does not seem to be affected by this added need for interpretation, or the visualization helps the user enough to outweigh the added task.

Occlusion of the pen tip already provides good results for *solid* objects in particular.

The result that no visualization technique clearly produces more accurate results in finding the starting position can be explained by the impact that occlusion of the pen tip already has on precision. All visualizations could use the pen tip to see when they penetrate the target object's surface.

For *wireframe* objects, the large confidence interval for *distanceToTarget* in the *minVis* condition indicates that the performance of participants differed a lot. On the other hand, the narrow confidence intervals of *heatmap* indicate that it could lead to accurate interaction more reliably.

In other projects investigating depth cues, using shadows achieved the best results for egocentric depth estimation with stereoscopic systems [e.g., Diaz et al., 2017; Hubona et al., 1999]. In our study, *heatmap* achieved better results than *shadow* particularly for subjective ratings and rankings of our participants. This is closer to the results by Wither and Höllerer [2005], since they state that in their study the color condition was more preferred than their shadow planes but the performance was largely the same. While not significant, participants in their study needed the most time for the shadow condition. This is similar to the trend in our findings that it took participants longer to reach the target when using the *shadow* visualization for *solid* objects.

The longer duration for *shadow* also occurred in other studies.

In conclusion, *heatmap* seems to be the best visualization used in this study, followed by *depth ray* and *shadow*.

8.5 Summary & Future Work

Moving a mid-air pen to a specific location in Handheld Augmented Reality is hard, as perceptual issues make it difficult to estimate the correct distance to a virtual object. However, aligning the pen with a virtual object is an essential mid-air interaction when trying to connect virtual objects to their real environment, such as in mid-air modeling applications. In addition to a baseline condition with minimal visualization, we designed four visualization techniques to show the 3D position of the pen in relation to its virtual environment, and compared their performance. A heatmap visualization, which shades every object in the scene based on its distance to the pen tip, achieved good results and was most preferred by participants.

We designed and evaluated depth visualizations to move to a specific point on a virtual object. A *heatmap* visualization performs best overall.

For this study, we chose the basic task of aligning the pointing device with a virtual object in the scene to gather find-

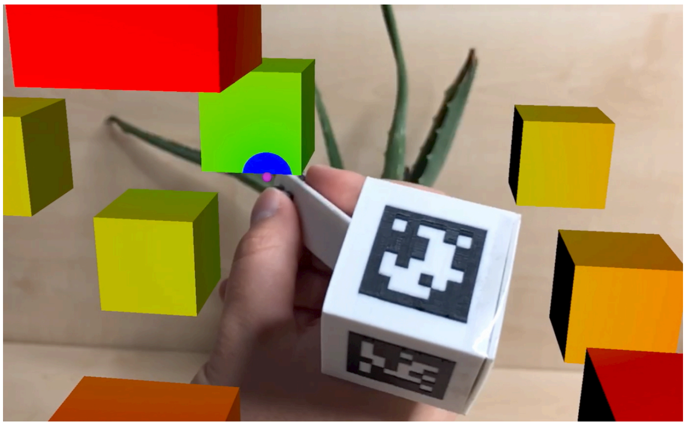


Figure 8.6: We used a linear gradient for the shading of objects in the *heatmap* technique.

ings that are applicable to many other settings. Further studies could look at more specific scenarios to see how specialized visualizations compare to the performance in the basic setup. For example, while our environment consisted of virtual objects, the inclusion of real world surfaces and objects is a promising direction to look at. Our study with 10 participants provides first insights into the effects of the different visualization techniques but studies with more users might uncover additional findings. Therefore, we provide our study software and data[1] to encourage replication and use of our results for further exploration.

Future studies could include real world objects.

The heatmap visualization adjusts the appearance of the whole scene instead of providing more localized feedback such as with the depth ray. Future studies could investigate this relationship closer to see whether different textures of objects and targets affect the targeting action and usage pattern of the visualization technique. Heatmap visualizations could also take textures into account and present the distance using different brightness or color adjustments that preserve texture structure. Our heatmap implementation used a computational linear gradient in RGB space which is not perceived as linear by humans (cf. Figure 8.6). Future studies could use more perceptually precise gradients [e.g., Lissner and Urban, 2012] to see whether this more accurate encoding of the distance between pen and envi-

Different color gradients could improve the heatmap visualization.

[1] hci.rwth-aachen.de/heatmaps (accessed 28.05.2021)

ronment improves the interaction even more. Furthermore, users with color-vision impairments might require specific color gradients to counterbalance their impairment. While the heatmap technique in this work encoded the Euclidean distance of the pen tip to objects regardless of the position of the camera, the color gradient could also be used to only encode the depth information from the current point of view. Since this is the most problematic dimension to perceive, this could focus the visualization on this dimension. However, movement of the camera would also impact the color of objects since it changes the depth axis, making it potentially more difficult to interpret the information.

Our work presented here looked at position visualizations for a single point in space. It would be interesting to see how rendering the real-time occlusion of virtual objects by the real pen and hand holding it affects the task of moving towards a specific location. While occluding objects behind the whole hand would certainly improve depth estimation overall, it would also hide more information in an already small window into the virtual environment. It seems like a promising research direction to investigate how visualizations from X-ray applications perform if applied to the *near* space and a more dynamic environment: In Handheld AR with a mid-air pen, the state between occluded and not occluded switches regularly for objects in the scene.

Occlusion of virtual objects with the hand and pen is an interesting research direction.

In our study, the task was to move to a specific location. However, for other interactions it may also be necessary to understand the overall layout of objects in a scene. While local visualizations like the depth ray are probably not the best techniques for such a task, global visualizations like the heatmap or shadows are promising candidates for further research. Unlike in our task, users would then have to comprehend and interpret different visualizations on objects to understand their relative positions to each other. Finally, scene composition also likely influences performance. In our study, we only showed eight cube objects for all visualizations to compare performance. Future studies could evaluate environments with varying object density, to understand how this affects performance of these visualization techniques.

Future work could investigate how visualizations affect scene understanding in general.

Having explored interactions and visualizations from all classes of mid-air interactions defined by LaViola et al. [2017], we sum up our findings and provide directions for future work in the next chapter.

Chapter 9

Summary and Future Work

Virtual environments such as Augmented Reality and Virtual Reality remove abstractions in the interaction with digital information. Instead of showing information in 2D on a screen, they place objects in 3D in the environment. In the case of Augmented Reality, this environment is to a large degree the real world and virtual objects can be shown as if they are really *in* the real world.

AR & VR place content in 3D around the user.

This opens up interesting interaction scenarios and one is *situated modeling* in which new virtual models are designed directly in the real world. Modeling directly in the real world allows seeing the new object immediately in context of the environment. An area where this is particularly important is Personal Fabrication, since objects fabricated using digital tools are often required to fit existing objects. *Situated modeling* allows designing these objects directly where they are meant to be used before printing them out. This also allows using the real object's physical properties during the design process to immediately attach the newly created object. For example, a user can model an inset for a cupholder in a car so that it can hold smaller sized cans. In this thesis, we explored elements of this interaction.

Situated modeling enables designers to create their models directly in the environment.

In this chapter, we first summarize the work described in this thesis (Chapter 9.1), highlighting our contributions.

In Chapter 9.2, we present options for further research in Handheld AR with a mid-air pen, before ending this thesis with closing remarks in Chapter 9.3.

9.1 Summary and Contributions

We presented an overview of AR.

After presenting definitions of Augmented Reality (AR) and a brief overview of the history of AR, we described the two major technological components of AR: *displays* are needed to show virtual objects embedded in the real world, and *tracking* systems are required to understand where the objects have to be shown. However, Augmented Reality has several issues with perceiving the scene. For example, studies have shown that it seems to be especially difficult to detect the correct depth position of virtual objects. Designers of interaction techniques in Augmented Reality need to keep these issues in mind.

Guidance types on objects can assist the user in performing a stroke.

Situated modeling in AR allows using the haptic properties of real world objects during the modeling process by tracing over the surfaces. Previous studies in VR have shown that having a flat physical surface improves the quality of drawn lines compared to those drawn without a physical surface. We investigated the drawing performance in Augmented Reality and under different surface conditions. For these surface conditions, we focussed on local changes that can guide a stroke on an object. We classified these into plain surfaces with *no guide*, surfaces with a *visual* line to follow, and changes in the shape of the surface to indicate a line along a *convex* or *concave* edge. These guidance types can occur both in physical as well as virtual objects.

Drawing on *physical* objects improves stroke accuracy.

To study their influence on the drawing performance, we combined an *optical see-through* display with *inside-out* tracking—a Microsoft HoloLens—with a Vicon *outside-in* tracking system, and let users draw around example objects featuring these guidance types. Throughout our measurements, we saw that drawing on *physical* objects compared to *virtual* objects improves the sketching accuracy the most. Strokes on *virtual* objects deviated on average around twice as much. The other guidance types also help to reduce the deviation compared to having *no guide*, except for *virtual* objects with a *concave* shape, for which the deviation does not improve. Another interesting finding for *virtual* objects is that while participants were able to keep the height of a target line consistent regardless of the guidance types, they consistently misjudged the position of the guide if one was present. This is likely a result of the issue of depth perception in virtual environments. Another part where this issue occurs is when comparing strokes drawn on the front of objects compared to those drawn on the back of objects. On *virtual* objects, strokes on the front are consistently drawn outside of the object, while strokes on the back also deviate inside the object. As a take-away from the study on physical guides, we calculated distances around the target stroke that contain 95 % of the drawn points to give an indication on tolerances for stroke optimization algorithms. The main distinction is between *physical* and *virtual* objects, and we calculated that the threshold for *virtual* objects needs to be about twice the size of the one for *physical* objects (*virtual*: 22.87 mm, *physical*: 11.15 mm).

The threshold for *virtual* objects is about twice the size of the one for *physical* objects.

However, the system we used for this study required expensive equipment and extensive calibration. To bring the possibilities of *situated modeling* to a broader audience, we need to consider technologies that are already in the hands of the users. At the moment, Handheld AR is the most used form of AR since it is supported on all current smartphones. To enable mid-air 3D pen input for Handheld AR, we developed the *ARPen system*. The ARPen uses marker tracking to calculate the position of a 3D-printed pen—or even just a marker on a business card. In an initial study, we determined that the best grip while holding the device is to rest it on the pinkie in landscape orientation, gripping it with the index finger. This also leaves the thumb free to touch the left side of the screen for touchscreen interaction.

We developed the ARPen system to enable *situated modeling* using only a smartphone and a 3D-printed pen.

The ARPen system is implemented with a basic plugin system so that the position of the mid-air pen can be interpreted in different ways. Since this simplifies the implementation of further interaction techniques for the system, we made the code of the ARPen system available open-source on Github[1] to enable other researchers to implement

The ARPen system is available open-source.

[1] github.com/i10/ARPen (accessed 28.05.2021)

and evaluate their interaction techniques for Handheld AR with a mid-air pen. A stable version of the ARPen is also available in the iOS App Store[2] to demonstrate the different interaction techniques. We presented the ARPen system at CHI'19 and the audience at the conference could download the App directly from the store and test out the interactions with paper markers we handed out. Between its release in April 2019 and writing this thesis in May 2021, the ARPen iOS App has been downloaded by over 1000 people. Over the course of a master's thesis, the ARPen system was extended with a modeling kernel that allows for the creation of complex geometries. With these capabilities, it is already possible to design objects directly in AR.

Many basic interaction techniques were not defined for such a system. We focussed on designing and evaluating these basic interaction techniques.

However, many objects require the combination of multiple components. For example, boolean operations, such as merging or intersection, require the arrangement of two objects and a subsequent combination. But, it was not directly clear how simple tasks such as selecting an object or moving it to a different position, are best achieved in Handheld AR with a mid-air pen. Therefore, we focussed on evaluating essential interaction tasks to build a solid understanding of the basic interactions. We structured these evaluations based on the classification by LaViola et al. [2017] who separated interaction in 3D user interfaces into *selection & manipulation*, *system control*, and *navigation*.

Casting a ray through the tip of the pen achieves the best results and is preferred by users.

For *selection & manipulation*, we designed and evaluated different techniques to *select* and *translate* virtual objects in the ARPen system. These techniques included isomorphic techniques, e.g., to drag and drop a virtual object as well as non-isomorphic ray-casting techniques using the touchscreen or mid-air pen. The ray-casting techniques, especially shooting a ray from the camera through the tip of the pen or using two-handed techniques on the touchscreen, achieved good results and were also preferred by the participants. However, participants also mentioned that the understanding of the spatial relationships of objects in the scene is reduced in the non-isomorphic techniques compared to the isomorphic techniques. Aside from the success of ray-casting techniques, another important factor of the interaction are ergonomic issues of keeping both arms

[2] hci.rwth-aachen.de/arpen-ios (accessed 28.05.2021)

Figure 9.1: With the ARPen system, the user can specify volumetric objects by tracing over physical objects (*left*). With our *selection & manipulation* techniques, the objects can be arranged relative to each other (*middle*) to be combined into more complex objects. The resulting model can be printed with a 3D printer (*right*).

raised for a longer period of time. This led to high ratings of the *touch&pen* technique to move an object to a new position because it enabled users to rest their hand with the pen on the table for most of the interaction—however, this was mostly possible because of our study setup. We also designed interaction techniques for *rotation* and *scaling* of virtual objects, which have been evaluated in bachelor's theses. With regard to *rotation*, the evaluation showed that direct techniques that apply the rotation of either the pen or the handheld device directly on the virtual object were preferred to pedal techniques that control the rate of rotation. For the scaling in mid-air, the touchscreen techniques *pinch* and *scrolling*, as well as *pen ray* achieved good results particularly in the subjective ratings of the participants. With these interaction techniques and the modeling capabilities of the ARPen system, it is possible to design the cup holder inset described in the introduction (cf. Figure 9.1).

For graphical menus, the *two-handed touch* menu performs best followed by *mid-air pen* & *one-handed touch*.

For the category of *system control*, we looked at the performance of different designs for context menus. Since context menus do not have to be attached to a mid-air position, we also considered techniques that fix the menu to the touchscreen or to a specific place in the real world so that haptic feedback from the surface can be used. However, we found that the haptic assistance of the surface does not overcome the drawbacks of a world-centered menu that requires additional movement and 3D position estimation. On the other hand, fixing the menu to the screen achieved good results, especially for picking an item from the menu. Overall, the two-handed menu achieved the best results.

However, except for the world-fixed *surface* menu, the other menu styles were not far behind, ergo the "best" technique will depend on the usage context around the menu. We presented potential use cases in which each menu type might be the best choice. For our scenario of mid-air modeling, for example, the benefits of the two-handed touchscreen menu might not be worth the trouble to continuously switch between input in mid-air and touch input on the screen.

We designed depth visualization and found that a *heatmap* visualization achieves good results overall.

According to LaViola et al. [2017], the category of *navigation* consists of two parts, the motoric component of moving to a new position, and the cognitive component of understanding the environment and determining *how* to move to a new position. This last component, called *wayfinding*, is particularly important for *situated modeling* in Handheld AR because of perception issues of detecting the correct depth of objects in the scene, an issue we have already encountered in the previous studies. Reducing the impact of this problem is why ray-casting approaches performed so well. To address this, we designed different depth visualizations showing the position of the pen tip relative to other objects in the scene. The *heatmap* visualization, which colors all objects in the scene based on their distance to the pen tip, achieved the best results particularly in the ratings by the participants. Our study also showed the importance of occlusion in determining the correct depth ordering and future studies should further evaluate how occlusion can be used in Handheld AR to improve the depth perception.

We provide further directions for future work in the following chapter.

9.2 Future Work

Our evaluations cover a small part of the interaction space.

The research of interaction techniques for mid-air pen input in Handheld AR presented in this thesis covers only a small part of this interaction space. In this chapter, we present open questions and further research possibilities. We used the categorization from LaViola et al. [2017] to address tasks that lead to solid foundational knowledge about the interaction in Handheld AR with a mid-air pen. The *Future Work* sections in each study chapter in this thesis contain further research directions based on the setup of each study. However, for each of the categories, there are further questions to be asked.

The individual *manipulation* techniques can be combined to create a consistent set of interactions.

For *selection & manipulation*, we have looked at each of the essential manipulations *translation*, *rotation*, and *scaling* individually so far. However, combining these into *compound actions*, e.g., to move and rotate an object simultaneously, is a logical next step in this category. Some of the interaction techniques we evaluated can be combined easily. For example, moving and rotating can be achieved simultaneously through the *pen ray pickup* and *direct pen* techniques. However, other techniques, such as *touchscreen* for rotation and *scrolling* for scaling, overlap or, in the case of *direct device* for rotation, limit the applicability of other techniques. Future projects should evaluate, how these actions can be combined intuitively and consistently. Potential solutions should either utilize individual manipulations modes applying the "best" individual manipulation technique or have a single manipulation mode in which each transformation type is controlled with a method distinctly different from the techniques of the other types—but not necessarily the "best" technique.

Constraints and *snapping* could improve interaction accuracy.

Questions that combine the *manipulation* and the *system control* categories are about *constraints* and *snapping*. *Constraints* limit, for example, which dimensions are affected by a manipulation. For a translation action, this could mean that an object is only moving along one axis, even though the input also contains changes of the remaining two axes. *Snapping* in the context of a translation could result in a movement that is not applied continuously but, for example, only positions the object on an application wide grid. The VR Multiplanes system by Machuca et al. [2018] uses snapping targets in the environment to enable the user to create straight lines or right angles. In Augmented Reality, these snapping targets could also be physical objects in the scene. How such modifications can be triggered and controlled in Handheld AR with a mid-air pen, has not been answered yet. Topics that solely fall into the *system control* category involve the state of the whole application. A potential for investigation is how pen-based gestures can be used to trigger actions. For example, "undo" and "redo" are currently triggered through buttons on the screen with the hand holding the pen. With global gestures, these actions could be triggered while keeping the hand holding the pen in mid-air.

Investigating the effects of and interaction with occlusion is a promising research direction.

An important aspect of *wayfinding* in the *navigation* category is occlusion of virtual object by real world objects. In Handheld AR with a mid-air pen, the hand holding the pen is frequently obstructing virtual information. However, current systems can not track real world occlusions at runtime but with added sensors and computation, this will become possible in the near future [e.g., Du et al., 2020]. At that time, it will be an interesting question how the occlusion affects the interaction with the system. On the one hand, the added occlusion will likely improve the understanding of depth ordering relative to the hand and pen, but, on the other hand, a large part of the small viewport could then be obstructed by the hand, hiding the virtual information behind it. How X-ray visualizations can be used in such a scenario, is an interesting question to research. Related to the *travel* component of *navigation*, future work could look more closely into ways of using the *camera-in-hand* metaphor of Handheld AR in combination with the mid-air pen. Since the space of directly interacting with the pen is limited to the area within the user's arms reach, moving the camera around can show the virtual information from a wide range of perspectives without requiring extensive movement of the user. However, in our studies, participants did not continuously use this option to adjust the viewing direction. Benscheid [2019] observed in his study of modeling techniques that participants adjusted the viewing angle in less than 8 % of trials. One participant commented that she stopped the movement because she felt that she would achieve better and quicker results by relying on her proprioception. Future studies could investigate if and how the flexible viewing direction can be utilized in Handheld AR with a mid-air pen or if the perceptive improvements of *perspective rendering* techniques are more beneficial even though they remove this flexibility. Another interesting possibility related to this is to evaluate *non-isomorphic* techniques for the interaction with mid-air models. *Isomorphic* techniques are potentially preferred when interacting with physical objects

Non-isomorphic modeling techniques could improve performance if a physical surface is not required.

because they allow using the haptic properties of these objects. However, as soon as the physical object is no longer required, e.g., because the outline has been traced, this benefit no longer exists and other interactions could be used. For example, the virtual model could be scaled up to allow for the drawing of finer details that would otherwise not be possible due to precision issues or hand shaking. The object would still keep a reference to the original size and position so that it could easily be placed back at the intended position again. Similarly, the representation on the handheld screen could also be adapted, for example, to show a zoomed-in portion of the screen and projecting the movements of the pen to that location to control a *virtual cursor*. This would allow seeing the context of the design while still enabling finer input. On the other hand, this might increase issues of keeping the markers of the pen inside the frustum of the camera, since the boundaries are not as visible.

Future work could focus on more specific modeling scenarios and tasks.

Our evaluation of interaction techniques so far have covered basic interaction tasks in isolation. Future work could build up on this foundation and focus on more specific usage scenarios. Benscheid [2019] already conducted the first study comparing basic modeling techniques. Future studies could, on the one hand, look more closely at techniques to achieve certain modeling actions, such as dynamic objects, and, on the other hand, the system can be used as a functional research prototype to study *situated modeling* scenarios outside of the lab. For example, the ARPen system could be used by participants over a longer period of time in which they have to complete modeling tasks and use the system. These experiences could be evaluated to uncover problems and potential solutions for *situated modeling* in Handheld AR. This can, for example, include questions about interaction techniques to attach a virtual object to a physical object. If the scene scanning capabilities of Handheld devices increase, not only occlusion calculation as mentioned above becomes possible but also the option to use the scan of the environment as input. In the use case of our cup holder, the system could then automatically scan the surfaces of the cup holder and the can, and the pen is used to refine the objects and specify their arrangement. The Mix&Match system by Stemasov et al. [2020] provides a first step into this direction for a head-mounted AR setup.

Apps like the ARPen could be used to collaborate and communicate 3D ideas.

Another potential of Handheld AR with a mid-air pen is in collaboration and communication. Similar to sketching an idea on paper and passing it around in a meeting, a system such as the ARPen could be used to enable this for 3D sketches. Multiple users could look into the same scene on their own device and share annotations and geometries. For example, a design company could place a prototype of a new product on the meeting room table and everyone is able to communicate their comments by directly sketching and annotating on the model for everyone to see. Research projects could investigate how the use of such a tool affects the communication and collaboration process [e.g., Wells and Houben, 2020; Villanueva et al., 2020]. This includes the interaction in remote working scenarios in which the object and pen actions are shared over longer distances. The mobility and simplicity of interaction in Handheld AR with a mid-air pen could also be beneficial for the use as a prototyping tool. Being able to quickly visualize an AR application allows exploration of different designs early in the process [Leiva et al., 2020]. With a system such as the ARPen, a user could sketch interfaces directly in mid-air to demonstrate the interaction with an AR application. Another application area could be to quickly arrange components of a movie set to plan shots at the *previsualization* stage [Volkmar et al., 2020].

Handheld AR with a mid-air pen could be used to prototype HMD applications to a degree.

In general, comparing the experience of interacting in Handheld AR with a mid-air pen to interacting in a head-mounted AR environment, e.g, by comparing questionnaire results as in [Putze et al., 2020], could indicate to what degree a system such as the ARPen could be used to quickly prototype interactions for HMD applications.

9.3 Closing Remarks

The *ultimate display* is still far away.

How will the future of Augmented Reality look like and how will the work presented in this thesis fit into this future? In the *ultimate display* by Sutherland [1965], the virtual and the real world are seamlessly merged and there is no handheld device required to see and interact with the virtual world. However, this vision is still quite far from reality. Studies since the time Sutherland wrote his essay have shown that interacting with the virtual world in the same way as with the real world might not be the optimal way, since it loses the "magic" possibilities that manipulating the virtual has.

We expect Handheld AR to remain the most prominent form of AR for the foreseeable future.

In the near future, head-mounted displays will be the closest we come to the vision of the *ultimate display*. However, current head-mounted, optical see-through displays have a narrow field-of-view and have not yet reached the consumer market. Recently, rumors have suggested that Apple is progressing in the development of AR glasses which could be a more consumer oriented device.[3] The adoption of such new devices will then also depend on the direct benefit to the user purchasing new hardware. Schmalstieg and Höllerer [2016] state that "AR for consumers must be a strict software-only solution delivered to devices the users already own" [p. 411]. Another component will be *social acceptability* of wearing a head-mounted device. Studies have shown that for the time being, especially people seeing a person with an HMD are critical of such a device [Koelle et al., 2015]. Based on these factors, we expect that Handheld AR will remain the most prominent form of Augmented Reality for the foreseeable future.

The combination of touchscreen and mid-air input has potential and could be used together with an HMD.

However, it is certain that future HMDs will improve aspects such as field-of-view, size, and weight, and that it will become more common to use such a device in public similar to the common usage of mobile phones and smartphones. How will our results fit into this development? Our studies show that the combination of touchscreen and mid-air input has a lot of potential and that, for example, in the case of menus, interacting with the touchscreen was the preferred way of interaction of our participants. Other projects also use the benefits of a touchscreen by including them in the immersive environment, for example, to improve the precision of interaction [Arora et al., 2018] or to offer additional interaction possibilities in combination with the head-mounted display [F. Zhu and Grossman, 2020]. See-through elements in an immersive environment have also been used to control the appearance of elements behind

[3] www.macrumors.com/2021/01/06/apple-glasses-prototyping-report/ (accessed 28.01.2021)

them [Schmalstieg et al., 1999]. So, a handheld touchscreen will still have a place for the interaction in Augmented Reality in the future and our evaluations of combining mid-air interaction and touchscreen can provide pointers on how to design this interaction.

We look forward to the technological advancements of AR in the future and are excited about the potential interactions Augmented Reality will make possible.

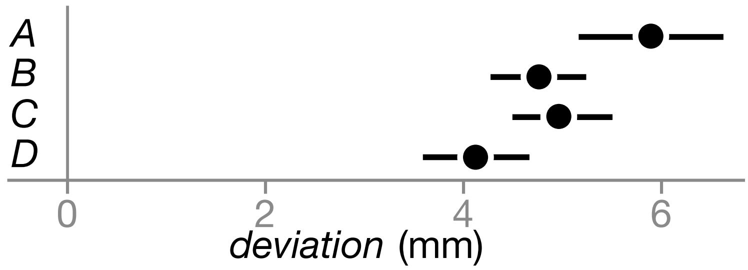
Appendix A

Evaluation with Confidence Intervals and Estimation

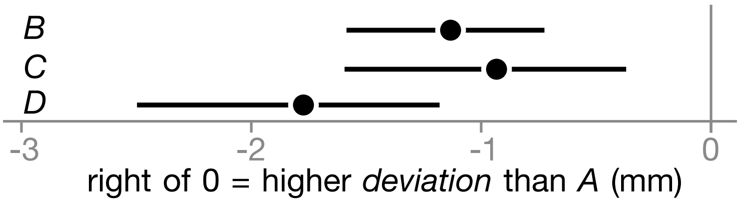
Over the recent years, the evaluation practices in HCI have been frequently criticised [e.g., Cockburn et al., 2020; Human–Computer Interaction Working Group, 2019; Dragicevic, 2016; Kaptein and Robertson, 2012]. A central point of critique is the reliance on null-hypothesis significance testing (NHST) and the dichotomous decision to accept a result as *significant* if the p-value is below 0.05. Instead, Dragicevic [2016], Cumming [2014], and others propose to communicate the uncertainty in the data by using confidence intervals and *estimation* of the effects.

Since these concepts are not as widespread as significance testing in the HCI community, we use this chapter to provide information on how we applied these concepts to the evaluation of our user studies and how to interpret the results. An in-depth overview of the issues with significance testing and justification for the analysis using confidence intervals and estimation is not the focus of this thesis so we refer to Dragicevic [2016] and Cumming [2014] as entry points for further information on this topic.

After running a user study, we aggregated the results for each condition per participant, for example, by averaging measurements or summing up successful selections. This left us with one measurement per condition per participant. We bootstrapped these aggregates with 1000 repetitions and calculated the 95 % confidence intervals using the BCa method [Carpenter and Bithell, 2000; Efron, 1987]. Plotting these intervals shows the overall results for a measurement and visualizes the uncertainty. We include these plots at the beginning of each results section. For example, the graph below shows a deviation measurement when using four different techniques A, B, C, and D.



However, for a more detailed evaluation and to be able to judge the effect of different conditions, we also calculate the within-subject effect. For this, we first specify a baseline condition that is either the current standard for interaction or is the minimal condition. For each aggregated measurement, we subtract the corresponding value of the baseline condition. This way, the results show the within-subject difference between the baseline condition and each other condition. We bootstrapped these differences with 1000 repetitions as well and also calculated the 95 % confidence intervals using the BCa method. For the graphs, that we are using in the main analysis section of our user studies, this means that values to the right of zero indicate that the baseline technique had less of the measured variable, while values to the left indicate higher values for the baseline compared with the other technique. Our analysis consists of interpreting the graphs and confidence intervals. This includes acknowledging the uncertainty of the data by avoiding dichotomous statements such as "D is better than B" while still pointing out trends in the data. For example, using the values that were used to create the graph above and setting technique A as the baseline, results in the graph below. The graph indicates that all other techniques (B, C, and D) seem to have less deviation compared to the baseline technique A. Additionally, it seems as if this effect is slightly larger for technique D while technique B and C seem more similar.



The evaluation in two previously published papers covered in this thesis have been performed using significance testing [Wacker et al., 2018b; Wacker et al., 2019]. To use a consistent style of evaluation, we have repeated the analysis of the user studies using confidence intervals and estimation as presented here.

Bibliography

Apple (2020). *Apple Unveils New iPad Pro with Breakthrough LiDAR Scanner and Brings Trackpad Support to iPadOS*. Accessed: 28.05.2021. URL: https://www.apple.com/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/.

Argelaguet, Ferran and Carlos Andujar (2013). "A Survey of 3D Object Selection Techniques for Virtual Environments". *Computers & Graphics* 37.3, pp. 121–136. ISSN: 0097-8493. DOI: 10.1016/j.cag.2012.12.003.

Armbrüster, Claudia, Marc Wolter, Torsten Kuhlen, Wilhelmus Spijkers, and Bruno Fimm (2008). "Depth Perception in Virtual Reality: Distance Estimations in Peri- and Extrapersonal Space". *CyberPsychology & Behavior* 11.1, pp. 9–15. DOI: 10.1089/cpb.2007.9935.

Arora, Rahul, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh (2018). "SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ". *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 185:1–185:15. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173759.

Arora, Rahul, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice (2017). "Experimental Evaluation of Sketching on Surfaces in VR". *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, pp. 5643–5654. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025474.

Azuma, Ronald T. (1997). "A Survey of Augmented Reality". *Presence: Teleoperators and Virtual Environments* 6.4, pp. 355–385. DOI: 10.1162/pres.1997.6.4.355.

Babic, Teo, Harald Reiterer, and Michael Haller (2018). "Pocket6: A 6DoF Controller Based On A Simple Smartphone Application". *Proceedings of the Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: ACM, pp. 2–10. ISBN: 978-1-4503-5708-1. DOI: 10.1145/3267782.3267785.

Bae, Seok-Hyung, Ravin Balakrishnan, and Karan Singh (2008). "ILoveSketch: As-Natural-As-Possible Sketching System for Creating 3D Curve Models". *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology - UIST '08*. ACM Press. DOI: 10.1145/1449715.1449740.

Bai, Huidong, Gun A. Lee, and Mark Billinghurst (2012). "Freeze View Touch and Finger Gesture Based Interaction Methods for Handheld Augmented Reality Interfaces". *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*. IVCNZ '12. Dunedin, New Zealand: ACM, pp. 126–131. ISBN: 978-1-4503-1473-2. DOI: 10.1145/2425836.2425864.

Bai, Huidong, Gun A. Lee, Mukundan Ramakrishnan, and Mark Billinghurst (2014). "3D Gesture Interaction for Handheld Augmented Reality". *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*. SA '14. Shenzhen, China: ACM, 7:1–7:6. ISBN: 978-1-4503-1891-4. DOI: 10.1145/2669062.2669073.

Benscheid, Jan (2019). "Making Sense of Lines: Interaction Sequences for 3D Modeling with Mid-Air Sketches". *Master's Thesis*.

Billinghurst, Mark, Tham Piumsomboon, and Huidong Bai (2014). "Hands in Space: Gesture Interaction with Augmented-Reality Interfaces". *IEEE Computer Graphics* *and Applications* 34.1, pp. 77–80. ISSN: 0272-1716. DOI: 10.1109/MCG.2014.8.

Brooks, Frederick P. (1999). "What's Real About Virtual Reality?" *IEEE Computer Graphics and Applications* 19.6, pp. 16–27. DOI: 10.1109/38.799723.

Butterworth, Jeff, Andrew Davidson, Stephen Hench, and Marc. T. Olano (1992). "3DM: A Three Dimensional Modeler Using a Head-mounted Display". *Proceedings of the 1992 Symposium on Interactive 3D Graphics*. I3D '92. Cambridge, Massachusetts, USA: ACM, pp. 135–138. ISBN: 0-89791-467-8. DOI: 10.1145/147156.147182.

Callahan, Jack, Don Hopkins, Mark Weiser, and Ben Shneiderman (1988). "An Empirical Comparison of Pie vs. Linear Menus". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA: ACM, pp. 95–100. ISBN: 0-201-14237-6. DOI: 10.1145/57167.57182.

Cao, Xiang and Shumin Zhai (2007). "Modeling Human Performance of Pen Stroke Gestures". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*. ACM Press. DOI: 10.1145/1240624.1240850.

Carpenter, James and John Bithell (2000). "Bootstrap Confidence Intervals: When, Which, What? A Practical Guide for Medical Statisticians". *Statistics in Medicine* 19.9, pp. 1141–1164. DOI: 10.1002/(SICI)1097-0258(20000515)19:9⟨1141::AID-SIM479⟩3.0.CO;2-F.

Caudell, Thomas P. and David W. Mizell (1992). "Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes". *Hawaii International Conference on System Sciences*, pp. 659–669.

Cipiloglu, Zeynep, Abdullah Bulbul, and Tolga Capin (2010). "A Framework for Enhancing Depth Perception in Computer Graphics". *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*. APGV

'10. Los Angeles, California: ACM, pp. 141–148. ISBN: 978-1-4503-0248-7. DOI: 10.1145/1836248.1836276.

Cockburn, Andy, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin (2020). "Threats of a Replication Crisis in Empirical Computer Science". *Commun. ACM* 63.8, pp. 70–79. ISSN: 0001-0782. DOI: 10.1145/3360311.

Cohé, Aurélie and Martin Hachet (2012). "Understanding User Gestures for Manipulating 3D Objects from Touch-screen Inputs". *Proceedings of Graphics Interface 2012*. GI '12. Toronto, Ontario, Canada: Canadian Information Processing Society, pp. 157–164. ISBN: 9781450314206.

Čopič Pucihar, Klen, Paul Coulton, and Jason Alexander (2013). "Evaluating Dual-view Perceptual Issues in Handheld Augmented Reality: Device vs. User Perspective Rendering". *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*. ICMI '13. Sydney, Australia: ACM, pp. 381–388. ISBN: 978-1-4503-2129-7. DOI: 10.1145/2522848.2522885.

Cosgrove, Ben (2012). *Pablo Picasso Draws With Light: The Story Behind an Iconic Photo*. Accessed: 28.05.2021. URL: https://www.life.com/arts-entertainment/behind-the-picture-picasso-draws-with-light/.

Cumming, Geoff (2014). "The New Statistics: Why and How". *Psychological Science* 25.1, pp. 7–29. DOI: 10.1177/0956797613504966.

Cutting, James E. (2003). "Reconceiving Perceptual Space". *Looking Into Pictures: An Interdisciplinary Approach to Pictorial Space*. Ed. by Heiko Hecht, Robert Schwartz, and Margaret Atherton. MIT Press, pp. 215–238.

Cutting, James E. and Peter M. Vishton (1995). "Perceiving Layout and Knowing Distances: The Integration, Relative Potency, and Contextual Use of Different Information about Depth". *Perception of Space and Motion*. Ed. by William Epstein and Sheena Rogers. Handbook of Perception and Cognition. San Diego: Academic Press, pp. 69–117. ISBN: 978-0-12-240530-3. DOI: 10.1016/B978-012240530-3/50005-5.

Dachselt, Raimund and Anett Hübner (2007). "Virtual Environments: Three-dimensional Menus: A Survey and Taxonomy". *Comput. Graph.* 31.1, pp. 53–65. ISSN: 0097-8493. DOI: 10.1016/j.cag.2006.09.006.

Dang, Nguyen-Thong and Daniel Mestre (2011). "Effects of Menu Orientation on Pointing Behavior in Virtual Environments". *Proceedings of the 2011 International Conference on Virtual and Mixed Reality: New Trends - Volume Part I.* Orlando, FL, USA: Springer-Verlag, pp. 144–153. ISBN: 978-3-642-22020-3.

Das, Kaushik and Christoph W. Borst (2010). "An Evaluation of Menu Properties and Pointing Techniques in a Projection-based VR Environment". *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. New York, NY, USA: IEEE, pp. 47–50. DOI: 10.1109/3DUI.2010.5444721.

De Araùjo, Bruno R., Géry Casiez, and Joaquim A. Jorge (2012). "Mockup Builder: Direct 3D Modeling on and Above the Surface in a Continuous Interaction Space". *Proceedings of Graphics Interface 2012*. GI '12. Toronto, Ontario, Canada: Canadian Information Processing Society, pp. 173–180. ISBN: 978-1-4503-1420-6.

Deering, Michael F. (1995). "HoloSketch: A Virtual Reality Sketching/Animation Tool". *ACM Trans. Comput.-Hum. Interact.* 2.3, pp. 220–238. ISSN: 1073-0516. DOI: 10.1145/210079.210087.

Dey, Arindam, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan (2018). "A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014". *Frontiers in Robotics and AI* 5, p. 37. ISSN: 2296-9144. DOI: 10.3389/frobt.2018.00037.

Dey, Arindam, Graeme Jarvis, Christian Sandor, and Gerhard Reitmayr (2012). "Tablet versus Phone: Depth Perception in Handheld Augmented Reality". *2012 IEEE International Symposium on Mixed and Augmented Reality (IS-*

MAR). New York, NY, USA: IEEE, pp. 187–196. DOI: 10.1109/ISMAR.2012.6402556.

Dey, Arindam and Christian Sandor (2014). "Lessons Learned: Evaluating Visualizations for Occluded Objects in Handheld Augmented Reality". *Int. J. Hum.-Comput. Stud.* 72.10-11, pp. 704–716. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2014.04.001.

Diaz, Catherine, Michael Walker, Danielle A. Szafir, and Daniel Szafir (2017). "Designing for Depth Perceptions in Augmented Reality". *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. New York, NY, USA: IEEE, pp. 111–122. DOI: 10.1109/ISMAR.2017.28.

Dragicevic, Pierre (2016). "Fair Statistical Communication in HCI". *Modern Statistical Methods for HCI*. Cham, Switzerland: Springer International Publishing, pp. 291–330. ISBN: 978-3-319-26633-6. DOI: 10.1007/978-3-319-26633-6_13.

Drascic, David and Paul Milgram (1996). "Perceptual Issues in Augmented Reality". *Proc. SPIE Vol. 2653: Stereoscopic Displays and Virtual Reality Systems III*, pp. 123–134. DOI: 10.1117/12.237425.

Drey, Tobias, Jan Gugenheimer, Julian Karlbauer, Maximilian Milo, and Enrico Rukzio (2020). "VRSketchIn: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual Reality". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–14. ISBN: 9781450367080. DOI: 10.1145/3313831.3376628.

Du, Ruofei, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim (2020). "DepthLab: Real-Time 3D Interaction with Depth Maps for Mobile Augmented Reality". *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, pp. 829–843. ISBN: 9781450375146. DOI: 10.1145/3379337.3415881.

Efron, Bradley (1987). "Better Bootstrap Confidence Intervals". *Journal of the American Statistical Association* 82.397, pp. 171–185. DOI: 10.1080/01621459.1987.10478410.

Ellis, Stephen R. and Brian M. Menges (1998). "Localization of Virtual Objects in the Near Visual Field". *Human Factors* 40.3, pp. 415–431. DOI: 10.1518/001872098779591278.

Engelbart, Douglas C. and William K. English (1968). "A Research Center for Augmenting Human Intellect". *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, pp. 395–410. ISBN: 9781450378994. DOI: 10.1145/1476589.1476645.

Fang, Cathy, Yang Zhang, Matthew Dworman, and Chris Harrison (2020). "Wireality: Enabling Complex Tangible Geometries in Virtual Reality with Worn Multi-String Haptics". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–10. ISBN: 9781450367080. DOI: 10.1145/3313831.3376470.

Feiner, Steven, Blair MacIntyre, Marcus Haupt, and Eliot Solomon (1993). "Windows on the World: 2D Windows for 3D Augmented Reality". *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. UIST '93. Atlanta, Georgia, USA: ACM, pp. 145–155. ISBN: 0-89791-628-X. DOI: 10.1145/168642.168657.

Fernquist, Jennifer, Tovi Grossman, and George Fitzmaurice (2011). "Sketch-Sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning". *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press. DOI: 10.1145/2047196.2047245.

Fitzmaurice, George W. (1993). "Situated Information Spaces and Spatially Aware Palmtop Computers". *Com-*

mun. ACM 36.7, pp. 39–49. ISSN: 0001-0782. DOI: 10.1145/159544.159566.

Flagg, Matthew and James M. Rehg (2006). “Projector-Guided Painting”. *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology - UIST '06*. ACM Press. DOI: 10.1145/1166253.1166290.

Fleisch, Timo, Gino Brunetti, Pedro Santos, and André Stork (2004). “Stroke-Input Methods for Immersive Styling Environments”. *Proceedings Shape Modeling Applications, 2004.* New York, NY, USA: IEEE. DOI: 10.1109/smi.2004.1314514.

Foley, James D., Andries van Dam, Steven K. Feiner, and John F. Hughes (1996). *Computer Graphics: Principles and Practice*. Addison-Wesley systems programming series. Addison-Wesley. ISBN: 9780201848403.

Gai, Wei, Cheng Lin, Chenglei Yang, Yulong Bian, Chia Shen, Xiangxu Meng, Lu Wang, Juan Liu, Mingda Dong, and Chengjie Niu (2017). “Supporting Easy Physical-to-Virtual Creation of Mobile VR Maze Games”. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press. DOI: 10.1145/3025453.3025494.

Gao, Zihan, Huiqiang Wang, Hongwu Lv, Moshu Wang, and Yifan Qi (2020). “Evaluating the Effects of Non-isomorphic Rotation on 3D Manipulation Tasks in Mixed Reality Simulation”. *To appear in IEEE Transactions on Visualization and Computer Graphics*. DOI: 10.1109/TVCG.2020.3010247.

Garrido-Jurado, Sergio, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez (2014). “Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion”. *Pattern Recognition* 47.6, pp. 2280–2292. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2014.01.005.

Gebhardt, Sascha, Sebastian Pick, Franziska Leithold, Bernd Hentschel, and Torsten Kuhlen (2013). “Extended

Pie Menus for Immersive Virtual Environments". *IEEE Transactions on Visualization and Computer Graphics* 19.4, pp. 644–651. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.31.

Gershenfeld, Neil (2007). *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. USA: Basic Books, Inc. ISBN: 0465027466.

Goh, Eg Su, Mohd Shahrizal Sunar, and Ajune Wanis Ismail (2019). "3D Object Manipulation Techniques in Handheld Mobile Augmented Reality Interface: A Review". *IEEE Access* 7, pp. 40581–40601. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2906394.

Gombač, Leo, Klen Čopič Pucihar, Matjaž Kljun, Paul Coulton, and Jan Grbac (2016). "3D Virtual Tracing and Depth Perception Problem on Mobile AR". *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. San Jose, California, USA: ACM, pp. 1849–1856. ISBN: 978-1-4503-4082-3. DOI: 10.1145/2851581.2892412.

Grimm, Cindy and Pushkar Joshi (2012). "Just DrawIt: A 3D Sketching System". *Proceedings of the International Symposium on Sketch-based Interfaces and Modeling*. SBIM '12. Annecy, France: Eurographics Association, pp. 121–130. ISBN: 978-3-905674-42-2.

Gustafson, Sean, Daniel Bierwirth, and Patrick Baudisch (2010). "Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback". *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*. UIST '10, pp. 3–12. DOI: 10.1145/1866029.1866033.

Harviainen, Tatu, Otto Korkalo, and Charles Woodward (2009). "Camera-based Interactions for Augmented Reality". *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*. ACE '09. Athens, Greece: Association for Computing Machinery, pp. 307–310. ISBN: 9781605588643. DOI: 10.1145/1690388.1690444.

Henrysson, Anders, Mark Billinghurst, and Mark Ollila (2005). "Virtual Object Manipulation Using a Mobile Phone". *Proceedings of the 2005 International Conference on Augmented Tele-existence*. ICAT '05. Christchurch, New Zealand: ACM, pp. 164–171. ISBN: 0-473-10657-4. DOI: 10.1145/1152399.1152430.

Henrysson, Anders, Joe Marshall, and Mark Billinghurst (2007). "Experiments in 3D Interaction for Mobile Phone AR". *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*. GRAPHITE '07. Perth, Australia: Association for Computing Machinery, pp. 187–194. ISBN: 9781595939128. DOI: 10.1145/1321261.1321295.

Huang, Hsin-Yu, Chih-Wei Ning, Po-Yao Wang, Jen-Hao Cheng, and Lung-Pan Cheng (2020). "Haptic-Go-Round: A Surrounding Platform for Encounter-Type Haptics in Virtual Reality Experiences". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–10. ISBN: 9781450367080. DOI: 10.1145/3313831.3376476.

Hubona, Geoffrey S., Philip N. Wheeler, Gregory W. Shirah, and Matthew Brandt (1999). "The Relative Contributions of Stereo, Lighting, and Background Scenes in Promoting 3D Depth Visualization". *ACM Trans. Comput.-Hum. Interact.* 6.3, pp. 214–242. ISSN: 1073-0516. DOI: 10.1145/329693.329695.

Human–Computer Interaction Working Group, Transparent Statistics in (2019). *Transparent Statistics Guidelines*. (Available at https://transparentstats.github.io/guidelines). DOI: 10.5281/zenodo.1186169.

Huo, Ke, Vinayak, and Karthik Ramani (2017). "Window-Shaping: 3D Design Ideation by Creating on, Borrowing from, and Looking at the Physical World". *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '17. Yokohama, Japan:

ACM, pp. 37–45. ISBN: 978-1-4503-4676-4. DOI: 10.1145/3024969.3024995.

Hürst, Wolfgang, Ronald Poppe, and Jerry van Angeren (2015). "Drawing Outside the Lines: Tracking-based Gesture Interaction in Mobile Augmented Entertainment". *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*, pp. 79–87.

Hürst, Wolfgang and Casper van Wezel (2013). "Gesture-based Interaction via Finger Tracking for Mobile Augmented Reality". *Multimedia Tools and Applications* 62.1, pp. 233–258. ISSN: 1573-7721. DOI: 10.1007/s11042-011-0983-y.

Iarussi, Emmanuel, Adrien Bousseau, and Theophanis Tsandilas (2013). "The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs". *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology - UIST '13*. ACM Press. DOI: 10.1145/2501988.2501997.

Igarashi, Takeo, Satoshi Matsuoka, and Hidehiko Tanaka (1999). "Teddy: A Sketching Interface for 3D Freeform Design". *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99*. ACM Press. DOI: 10.1145/311535.311602.

IKEA (2017). *IKEA Place App*. Accessed: 28.05.2021. URL: https://www.ikea.com/ch/en/customer-service/mobile-apps/ikea-place-app-pub0bab12b1.

Insko, Brent Edward (2001). "Passive Haptics Significantly Enhances Virtual Environments". PhD thesis. University of North Carolina at Chapel Hill.

Issartel, Paul, Florimond Guéniat, and Mehdi Ammi (2014). "Slicing Techniques for Handheld Augmented Reality". *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. New York, NY, USA: IEEE, pp. 39–42. DOI: 10.1109/3DUI.2014.6798839.

Jackson, Bret and Daniel F. Keefe (2011). "Sketching Over Props: Understanding and Interpreting 3D Sketch Input Relative to Rapid Prototype Props". *IUI '11 Sketch Recognition Workshop*. DOI: 10.1145/2047196.2047245.

Jackson, Bret and Daniel F. Keefe (2016). "Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR". *IEEE Transactions on Visualization and Computer Graphics* 22.4, pp. 1442–1451. DOI: 10.1109/tvcg.2016.2518099.

Jacoby, Richard H. and Stephen R. Ellis (1992). "Using Virtual Menus in a Virtual Environment". *Proceedings of SPIE - The International Society for Optical Engineering* 1668, pp. 39–48. DOI: 10.1117/12.59654.

Jones, J. Adam, J. Edward Swan, Gurjot Singh, Eric Kolstad, and Stephen R. Ellis (2008). "The Effects of Virtual Reality, Augmented Reality, and Motion Parallax on Egocentric Depth Perception". *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*. APGV '08. Los Angeles, California: ACM, pp. 9–14. ISBN: 978-1-59593-981-4. DOI: 10.1145/1394281.1394283.

Jurgens, Volkert, Andy Cockburn, and Mark Billinghurst (2006). "Depth Cues for Augmented Reality Stakeout". *Proceedings of the 7th ACM SIGCHI New Zealand Chapter's International Conference on Computer-human Interaction: Design Centered HCI*. CHINZ '06. Christchurch, New Zealand: ACM, pp. 117–124. ISBN: 1-59593-473-1. DOI: 10.1145/1152760.1152775.

Kang, Yuna, Hyungki Kim, Hiromasa Suzuki, and Soonhung Han (2015). "Editing 3D Models on Smart Devices". *Computer-Aided Design* 59, pp. 229–238. ISSN: 0010-4485. DOI: 10.1016/j.cad.2013.08.001.

Kaptein, Maurits and Judy Robertson (2012). "Rethinking Statistical Analysis Methods for CHI". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, pp. 1105–1114. ISBN: 9781450310154. DOI: 10.1145/2207676.2208557.

Kato, Hirokazu and Mark Billinghurst (1999). "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System". *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pp. 85–94. DOI: 10.1109/IWAR.1999.803809.

Keefe, Daniel F., Robert C. Zeleznik, and David H. Laidlaw (2007). "Drawing on Air: Input Techniques for Controlled 3D Line Illustration". *IEEE Transactions on Visualization and Computer Graphics* 13.5, pp. 1067–1081. DOI: 10.1109/tvcg.2007.1060.

Kiger, John I. (1984). "The Depth/Breadth Trade-off in the Design of Menu-driven User Interfaces". *Int. J. Man-Mach. Stud.* 20.2, pp. 201–213. ISSN: 0020-7373. DOI: 10.1016/S0020-7373(84)80018-8.

Kim, Namgyu, Gerard Jounghyun Kim, Chan-Mo Park, Inseok Lee, and Sung H. Lim (2000). "Multimodal Menu Presentation and Selection in Immersive Virtual Environments". *Proceedings of the IEEE Virtual Reality 2000 Conference*. VR '00. Washington, DC, USA: IEEE Computer Society. ISBN: 0-7695-0478-7.

Klamma, Donna (2019). "Rotating Objects: Implementation and Evaluation of Rotation Techniques for the ARPen System". *Bachelor's Thesis*.

Klein, Georg and David Murray (2007). "Parallel Tracking and Mapping for Small AR Workspaces". *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852.

Koelle, Marion, Matthias Kranz, and Andreas Möller (2015). "Don't Look at Me That Way! Understanding User Attitudes Towards Data Glasses Usage". *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '15. Copenhagen, Denmark: Association for Computing Machinery, pp. 362–372. ISBN: 9781450336529. DOI: 10.1145/2785830.2785842.

Komerska, Rick and Colin Ware (2004). “A Study of Haptic Linear and Pie Menus in a 3D Fish Tank VR Environment”. *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '04)*. New York, NY, USA: IEEE, pp. 224–231. DOI: 10.1109/HAPTIC.2004.1287200.

Kruijff, Ernst, J. Edward Swan, and Steven Feiner (2010). “Perceptual Issues in Augmented Reality Revisited”. *2010 IEEE International Symposium on Mixed and Augmented Reality*. New York, NY, USA: IEEE, pp. 3–12. DOI: 10.1109/ISMAR.2010.5643530.

Kruszyński, Krzysztof Jakub and Robert van Liere (2009). “Tangible Props for Scientific Visualization: Concept, Requirements, Application”. *Virtual Reality* 13.4, pp. 235–244. DOI: 10.1007/s10055-009-0126-1.

Lakatos, David, Matthew Blackshaw, Alex Olwal, Zachary Barryte, Ken Perlin, and Hiroshi Ishii (2014). “T(Ether): Spatially-aware Handhelds, Gestures and Proprioception for Multi-user 3D Modeling and Animation”. *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*. SUI '14. Honolulu, Hawaii, USA: ACM, pp. 90–93. ISBN: 978-1-4503-2820-3. DOI: 10.1145/2659766.2659785.

Landauer, T. K. and D. W. Nachbar (1985). “Selection from Alphabetic and Numeric Menu Trees Using a Touch Screen: Breadth, Depth, and Width”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '85. San Francisco, California, USA: ACM, pp. 73–78. ISBN: 0-89791-149-0. DOI: 10.1145/317456.317470.

Lau, Manfred, Greg Saul, Jun Mitani, and Takeo Igarashi (2010). “Modeling-in-Context: User Design of Complementary Objects with a Single Photo”. *Proceedings of the Seventh Sketch-based Interfaces and Modeling Symposium*. Eurographics Association, pp. 17–24. DOI: 10.2312/sbm/sbm10/017-024.

Laviola, Joseph J. and Michael Katzourin (2007). “An Exploration of Non-Isomorphic 3D Rotation in Surround

Screen Virtual Environments". *2007 IEEE Symposium on 3D User Interfaces*. DOI: 10.1109/3DUI.2007.340774.

LaViola, Joseph J., Ernst Kruijff, Ryan P. McMahan, Doug Bowman, and Ivan P. Poupyrev (2017). *3D User Interfaces: Theory and Practice*. en. 2nd edition. Boston: Addison-Wesley. ISBN: 978-0-13-403432-4.

Laviole, Jérémy and Martin Hachet (2012). "PapARt: Interactive 3D Graphics and Multi-Touch Augmented Paper for Artistic Creation". *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. New York, NY, USA: IEEE, pp. 3–6. DOI: 10.1109/3dui.2012.6184167.

Lee, Gun A., Ungyeon Yang, Yongwan Kim, Dongsik Jo, Ki-Hong Kim, Jae Ha Kim, and Jin Sung Choi (2009). "Freeze-Set-Go Interaction Method for Handheld Mobile Augmented Reality Environments". *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. VRST '09. Kyoto, Japan: ACM, pp. 143–146. ISBN: 978-1-60558-869-8. DOI: 10.1145/1643928.1643961.

Lee, Hyeongmook, Dongchul Kim, and Woontack Woo (2011). "Graphical Menus Using a Mobile Phone for Wearable AR Systems". *Proceedings of the 2011 International Symposium on Ubiquitous Virtual Reality*. ISUVR '11. Washington, DC, USA: IEEE Computer Society, pp. 55–58. ISBN: 978-0-7695-4420-5. DOI: 10.1109/ISUVR.2011.23.

Lee, Joon Hyub, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae (2018). "Projective Windows: Bringing Windows in Space to the Fingertip". *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 218:1–218:8. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173792.

Leiva, Germán, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente (2020). "Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376160.

Lissner, Ingmar and Philipp Urban (2012). “Toward a Unified Color Space for Perception-based Image Processing”. *IEEE Transactions on Image Processing* 21.3, pp. 1153–1168. ISSN: 1941-0042. DOI: 10.1109/TIP.2011.2163522.

Livingston, Mark A., Zhuming Ai, J. Edward Swan, and Harvey S. Smallman (2009). “Indoor vs. Outdoor Depth Perception for Mobile Augmented Reality”. *2009 IEEE Virtual Reality Conference*. New York, NY, USA: IEEE, pp. 55–62. DOI: 10.1109/VR.2009.4810999.

Lopes, Pedro, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch (2017). “Providing Haptics to Walls & Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation”. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press. DOI: 10.1145/3025453.3025600.

Machuca, Mayra D. Barrera, Paul Asente, Wolfgang Stuerzlinger, Jingwan Lu, and Byungmoon Kim (2018). “Multiplanes: Assisted Freehand VR Sketching”. *Proceedings of the Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: Association for Computing Machinery, pp. 36–47. ISBN: 9781450357081. DOI: 10.1145/3267782.3267786.

Marzo, Asier, Benoît Bossavit, and Martin Hachet (2014). “Combining Multi-touch Input and Device Movement for 3D Manipulations in Mobile Augmented Reality Environments”. *Proceedings of the 2nd ACM Symposium on Spatial User Interaction*. SUI '14. Honolulu, Hawaii, USA: ACM, pp. 13–16. ISBN: 978-1-4503-2820-3. DOI: 10.1145/2659766.2659775.

McNeely, William A. (1993). “Robotic Graphics: A New Approach to Force Feedback for Virtual Reality”. *Proceedings of IEEE Virtual Reality Annual International Symposium*. New York, NY, USA: IEEE. DOI: 10.1109/vrais.1993.380761.

Microsoft (2015). *Microsoft HoloLens: Press Materials*. Accessed: 28.05.2021. URL: https://news.microsoft.com/microsoft-hololens-press-materials/.

Microsoft (2016). *Introducing First Ever Experiences for the Microsoft HoloLens Development Edition*. Accessed: 28.05.2021. URL: https://blogs.windows.com/devices/2016/02/29/introducing-first-ever-experiences-for-the-microsoft-hololens-development-edition/.

Milgram, Paul and Fumio Kishino (1994). "A Taxonomy of Mixed Reality Visual Displays". *IEICE Transactions on Information Systems* E77-D. URL: http://vered.rose.utoronto.ca/people/paul%5C_dir/IEICE94/ieice.html.

Miller, Dwight P. (1981). "The Depth/Breadth Tradeoff in Hierarchical Computer Menus". *Proceedings of the Human Factors Society Annual Meeting* 25.1, pp. 296–300. DOI: 10.1177/107118138102500179.

Milosevic, Bojan, Flavio Bertini, Elisabetta Farella, and Serena Morigi (2016). "A SmartPen for 3D Interaction and Sketch-based Surface Modeling". *The International Journal of Advanced Manufacturing Technology* 84.5, pp. 1625–1645. ISSN: 1433-3015. DOI: 10.1007/s00170-015-7828-1.

Mohammed, Farhadiba (2020). "Scaling Objects: Implementation and Evaluation of Scaling Techniques for the ARPen System". *Bachelor's Thesis*.

Mossel, Annette, Benjamin Venditti, and Hannes Kaufmann (2013a). "3DTouch and HOMER-S: Intuitive Manipulation Techniques for One-handed Handheld Augmented Reality". *Proceedings of the Virtual Reality International Conference: Laval Virtual*. VRIC '13. Laval, France: ACM, 12:1–12:10. ISBN: 978-1-4503-1875-4. DOI: 10.1145/2466816.2466829.

Mossel, Annette, Benjamin Venditti, and Hannes Kaufmann (2013b). "DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments". *Proceedings of the Virtual Reality International Conference: Laval Virtual*. VRIC '13. Laval, France: ACM, 10:1–10:10. ISBN: 978-1-4503-1875-4. DOI: 10.1145/2466816.2466827.

Mota, Catarina (2011). "The Rise of Personal Fabrication". *Proceedings of the 8th ACM Conference on Creativ-*

ity and Cognition. C&C '11. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 279–288. ISBN: 9781450308205. DOI: 10.1145/2069618.2069665.

Newcombe, Richard A., Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon (2011). "KinectFusion: Real-time Dense Surface Mapping and Tracking". *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136. DOI: 10.1109/ISMAR.2011.6092378.

Nielsen, Jakob (1993). *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0125184050.

Norman, Donald A. (2013). *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc. ISBN: 978-0465050659.

Nowak, Oliver (2019). "ARMenus: An Evaluation of Menu Interfaces in Pen-based AR Applications on Smartphones". *Master's Thesis*.

Olsen, Luke, Faramarz F. Samavati, Mario Costa Sousa, and Joaquim A. Jorge (2009). "Sketch-based Modeling: A Survey". *Computers & Graphics* 33.1, pp. 85–103. DOI: 10.1016/j.cag.2008.09.013.

Paoli, Chris De and Karan Singh (2015). "SecondSkin: Sketch-based Construction of Layered 3D Models". *ACM Transactions on Graphics* 34.4, 126:1–126:10. DOI: 10.1145/2766948.

Pastel, Robert (2006). "Measuring the Difficulty of Steering Through Corners". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Quebec, Canada: ACM, pp. 1087–1096. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124934.

Peddie, Jon (2017). *Augmented Reality - Where We Will All Live*. Cham, Switzerland: Springer International Publishing. ISBN: 978-3-319-54501-1. DOI: 10.1007/978-3-319-54502-8.

Peng, Huaishu, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière (2018). "RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer". *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 579:1–579:12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174153.

Polvi, Jarkko, Takafumi Taketomi, Goshiro Yamamoto, Arindam Dey, Christian Sandor, and Hirokazu Kato (2016). "SlidAR: A 3D Positioning Method for SLAM-based Handheld Augmented Reality". *Computers & Graphics* 55, pp. 33–43. ISSN: 0097-8493. DOI: 10.1016/j.cag.2015.10.013.

Poupyrev, Ivan, Suzanne Weghorst, and Sidney Fels (2000). "Non-Isomorphic 3D Rotational Techniques". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: Association for Computing Machinery, pp. 540–547. ISBN: 1581132166. DOI: 10.1145/332040.332497.

Putze, Susanne, Dmitry Alexandrovsky, Felix Putze, Sebastian Höffner, Jan David Smeddinck, and Rainer Malaka (2020). "Breaking The Experience: Effects of Questionnaires in VR User Studies". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–15. ISBN: 9781450367080. DOI: 10.1145/3313831.3376144.

Rekimoto, Jun and Katashi Nagao (1995). "The World through the Computer: Computer Augmented Interaction with Real World Environments". *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. UIST '95. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, pp. 29–36. ISBN: 089791709X. DOI: 10.1145/215585.215639.

Rivers, Alec, Andrew Adams, and Frédo Durand (2012). "Sculpting by Numbers". *ACM Transactions on Graphics* 31.6, 157:1–157:7. DOI: 10.1145/2366145.2366176.

Rosa, Nina, Remco C. Veltkamp, Wolfgang Hürst, Tanja Nijboer, Carolien Gilbers, and Peter Werkhoven (2019). "The Supernumerary Hand Illusion in Augmented Reality". *ACM Trans. Appl. Percept.* 16.2, 12:1–12:20. ISSN: 1544-3558. DOI: 10.1145/3341225.

Sachs, Emanuel, Andrew Roberts, and David Stoops (1991). "3-Draw: A Tool for Designing 3D Shapes". *IEEE Computer Graphics and Applications* 11.6, pp. 18–26. ISSN: 0272-1716. DOI: 10.1109/38.103389.

Schäfer, René (2020). "Redesigning ARPen: Evaluating Different Marker Placements for Mid-Air Pen Interaction". *Master's Thesis*.

Schmalstieg, Dieter, L. Miguel Encarnação, and Zsolt Szalavári (1999). "Using Transparent Props for Interaction with the Virtual Table". *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. I3D '99. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 147–153. ISBN: 1581130821. DOI: 10.1145/300523.300542.

Schmalstieg, Dieter and Tobias Höllerer (2016). *Augmented Reality - Principles and Practice*. English. United States: Addison-Wesley Professional. ISBN: 978-0321883575.

Schmid, Johannes, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner (2011). "OverCoat: An Implicit Canvas for 3D Painting". *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11*. ACM Press. DOI: 10.1145/1964921.1964923.

Schmidt, Ryan, Azam Khan, Karan Singh, and Gord Kurtenbach (2009). "Analytic Drawing of 3D Scaffolds". *ACM Transactions on Graphics* 28.5, 149:1–149:10. DOI: 10.1145/1618452.1618495.

Schwind, Valentin, Pascal Knierim, Lewis Chuang, and Niels Henze (2017). ""Where's Pinky?": The Effects of a

Reduced Number of Fingers in Virtual Reality". *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '17. Amsterdam, The Netherlands: ACM, pp. 507–515. ISBN: 978-1-4503-4898-0. DOI: 10.1145/3116595.3116596.

Seidinger, Iris and Jens Grubert (2016). "3D Character Customization for Non-Professional Users in Handheld Augmented Reality". *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. New York, NY, USA: IEEE, pp. 129–134. DOI: 10.1109/ISMAR-Adjunct.2016.0059.

Song, Hyunyoung, François Guimbretière, Chang Hu, and Hod Lipson (2006). "ModelCraft: Capturing Freehand Annotations and Edits on Physical 3D Models". *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology - UIST '06*. ACM Press. DOI: 10.1145/1166253.1166258.

Stemasov, Evgeny, Tobias Wagner, Jan Gugenheimer, and Enrico Rukzio (2020). "Mix&Match: Towards Omitting Modelling Through In-Situ Remixing of Model Repository Artifacts in Mixed Reality". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376839.

Surale, Hemant Bhaskar, Aakar Gupta, Mark Hancock, and Daniel Vogel (2019). "TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality". *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, pp. 1–13. ISBN: 9781450359702. DOI: 10.1145/3290605.3300243.

Sutherland, Ivan E. (1963). "Sketchpad: A Man-Machine Graphical Communication System". *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*. AFIPS '63 (Spring). Detroit, Michigan: Association for Computing Machinery, pp. 329–346. ISBN: 9781450378802. DOI: 10.1145/1461551.1461591.

Sutherland, Ivan E. (1965). “The Ultimate Display”. *Proceedings of the IFIP Congress*, pp. 506–508.

Sutherland, Ivan E. (1968). “A Head-Mounted Three Dimensional Display”. *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS ’68 (Fall, part I). San Francisco, California: Association for Computing Machinery, pp. 757–764. ISBN: 9781450378994. DOI: 10.1145/1476589.1476686.

Swan, J. Edward, Liisa Kuparinen, Scott Rapson, and Christian Sandor (2017). “Visually Perceived Distance Judgments: Tablet-based Augmented Reality Versus the Real World”. *International Journal of Human-Computer Interaction* 33.7, pp. 576–591. DOI: 10.1080/10447318.2016.1265783.

Swan, J. Edward, Mark A. Livingston, Harvey S. Smallman, Dennis Brown, Yohan Baillot, Joseph L. Gabbard, and Deborah Hix (2006). “A Perceptual Matching Technique for Depth Judgments in Optical, See-Through Augmented Reality”. *IEEE Virtual Reality Conference (VR 2006)*. New York, NY, USA: IEEE, pp. 19–26. DOI: 10.1109/VR.2006.13.

Tang, Xiao, Xiaowei Hu, Chi-Wing Fu, and Daniel Cohen-Or (2020). “GrabAR: Occlusion-Aware Grabbing Virtual Objects in AR”. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST ’20. Virtual Event, USA: Association for Computing Machinery, pp. 697–708. ISBN: 9781450375146. DOI: 10.1145/3379337.3415835.

Tsuda, Takahiro, Haruyoshi Yamamoto, Yoshinari Kameda, and Yuichi Ohta (2005). “Visualization Methods for Outdoor See-through Vision”. *Proceedings of the 2005 International Conference on Augmented Tele-existence*. ICAT ’05. Christchurch, New Zealand: ACM, pp. 62–69. ISBN: 0-473-10657-4. DOI: 10.1145/1152399.1152412.

van Teylingen, Ron, William Ribarsky, and Charles van der Mast (1997). “Virtual Data Visualizer”. *IEEE Trans-*

actions on Visualization and Computer Graphics 3.1, pp. 65–74. ISSN: 1077-2626. DOI: 10.1109/2945.582350.

Villanueva, Ana, Zhengzhe Zhu, Ziyi Liu, Kylie Peppler, Thomas Redick, and Karthik Ramani (2020). "Meta-AR-App: An Authoring Platform for Collaborative Augmented Reality in STEM Classrooms". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–14. ISBN: 9781450367080. DOI: 10.1145/3313831.3376146.

Vincent, Thomas, Laurence Nigay, and Takeshi Kurata (2013). "Precise Pointing Techniques for Handheld Augmented Reality". *Human-Computer Interaction – INTERACT 2013*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 122–139. ISBN: 978-3-642-40483-2.

Volkmar, Georg, Thomas Muender, Dirk Wenig, and Rainer Malaka (2020). "Evaluation of Natural User Interfaces in the Creative Industries". *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–8. ISBN: 9781450368193. DOI: 10.1145/3334480.3375201.

Wacker, Philipp, Oliver Nowak, Simon Voelker, and Jan Borchers (2019). "ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone". *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: ACM, 619:1–619:12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300849.

Wacker, Philipp, Oliver Nowak, Simon Voelker, and Jan Borchers (2020a). "Evaluating Menu Techniques for Handheld AR with a Smartphone & Mid-Air Pen". *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '20. Oldenburg, Germany: ACM. ISBN: 978-1-4503-7516-0/20/10. DOI: 10.1145/3290605.3300849.

Wacker, Philipp, Adrian Wagner, Simon Voelker, and Jan Borchers (2018a). “Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality”. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI EA ’18. Montreal QC, Canada: Association for Computing Machinery, pp. 1–6. ISBN: 9781450356213. DOI: 10.1145/3170427.3188493.

Wacker, Philipp, Adrian Wagner, Simon Voelker, and Jan Borchers (2018b). “Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality”. *Proceedings of the Symposium on Spatial User Interaction*. SUI ’18. Berlin, Germany: ACM, pp. 25–35. ISBN: 978-1-4503-5708-1. DOI: 10.1145/3267782.3267788.

Wacker, Philipp, Adrian Wagner, Simon Voelker, and Jan Borchers (2020b). “Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR”. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–11. ISBN: 9781450367080. DOI: 10.1145/3313831.3376848.

Wagner, Adrian (2018). “Analyzing 3D Sketching Performance on Physical Objects in Augmented Reality”. *Master’s Thesis*.

Wagner, Daniel, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg (2008). “Pose Tracking from Natural Features on Mobile Phones”. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 125–134. DOI: 10.1109/ISMAR.2008.4637338.

Wagner, Daniel and Dieter Schmalstieg (2003). “First Steps Towards Handheld Augmented Reality”. *Proceedings of the Seventh IEEE International Symposium on Wearable Computers, 2003.* IEEE, pp. 127–135.

Wanger, Leonard R., James A. Ferwerda, and Donald P. Greenberg (1992). “Perceiving Spatial Relationships in

Computer-generated Images". *IEEE Computer Graphics and Applications* 12.3, pp. 44–58. ISSN: 0272-1716. DOI: 10.1109/38.135913.

Wann, John P., Simon Rushton, and Mark Mon-Williams (1995). "Natural Problems for Stereoscopic Depth Perception in Virtual Environments". *Vision Research* 35.19, pp. 2731–2736. ISSN: 0042-6989. DOI: 10.1016/0042-6989(95)00018-U.

Wehnert, Felix (2018). "Pen-based Drawing in Augmented Reality on Mobile Phones". *Bachelor's Thesis*.

Weichel, Christian, Jason Alexander, Abhijit Karnik, and Hans Gellersen (2015). "SPATA: Spatio-Tangible Tools for Fabrication-Aware Design". *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14*. ACM Press. DOI: 10.1145/2677199.2680576.

Weichel, Christian, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen (2014). "MixFab: A Mixed-Reality Environment for Personal Fabrication". *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*. ACM Press. DOI: 10.1145/2556288.2557090.

Wells, Thomas and Steven Houben (2020). "CollabAR – Investigating the Mediating Role of Mobile AR Interfaces on Co-Located Group Collaboration". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376541.

Wesche, Gerold and Hans-Peter Seidel (2001). "FreeDrawer: A Free-form Sketching System on the Responsive Workbench". *Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST '01*. ACM Press. DOI: 10.1145/505008.505041.

Wiese, Eva, Johann Habakuk Israel, Achim Meyer, and Sara Bongartz (2010). "Investigating the Learnability of Immersive Free-hand Sketching". *Proceedings of the Seventh Sketch-based Interfaces and Modeling Symposium*. SBIM '10. Annecy, France: Eurographics Association, pp. 135–142. ISBN: 978-3-905674-25-5.

Wither, Jason and Tobias Höllerer (2005). "Pictorial Depth Cues for Outdoor Augmented Reality". *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. New York, NY, USA: IEEE, pp. 92–99. DOI: 10.1109/ISWC.2005.41.

Wu, Po-Chen, Robert Wang, Kenrick Kin, Christopher Twigg, Shangchen Han, Ming-Hsuan Yang, and Shao-Yi Chien (2017). "DodecaPen: Accurate 6DoF Tracking of a Passive Stylus". *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. Quebec City, QC, Canada: ACM, pp. 365–374. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126664.

Xin, Min, Ehud Sharlin, and Mario Costa Sousa (2008). "Napkin Sketch: Handheld Mixed Reality 3D Sketching". *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*. VRST '08. Bordeaux, France: Association for Computing Machinery, pp. 223–226. ISBN: 9781595939517. DOI: 10.1145/1450579.1450627.

Xu, Baoxuan, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh (2014). "True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization". *ACM Transactions on Graphics* 33.4, pp. 1–13. DOI: 10.1145/2601097.2601128.

Yee, Brandon, Yuan Ning, and Hod Lipson (2009). "Augmented Reality In-situ 3D Sketching of Physical Objects". *Intelligent UI Workshop on Sketch Recognition*. IUI '09.

Yoon, Sang Ho, Ke Huo, and Karthik Ramani (2016). "TMotion: Embedded 3D Mobile Input Using Magnetic Sensing Technique". *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '16. Eindhoven, Netherlands: ACM, pp. 21–29. ISBN: 978-1-4503-3582-9. DOI: 10.1145/2839462.2839463.

Yue, Ya-Ting, Xiaolong Zhang, Yongliang Yang, Gang Ren, Yi-King Choi, and Wenping Wang (2017). "WireDraw: 3D Wire Sculpturing Guided with Mixed Reality". *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press. DOI: 10.1145/3025453.3025792.

Zhu, Fengyuan and Tovi Grossman (2020). "BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–14. ISBN: 9781450367080. DOI: 10.1145/3313831.3376233.

Zhu, Kening, Alexandru Dancu, and Shengdong (Shen) Zhao (2016). "FusePrint: A DIY 2.5D Printing Technique Embracing Everyday Artifacts". *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16*. ACM Press. DOI: 10.1145/2901790.2901792.

Own Publications

Papers (Peer-reviewed, archival)

Hueber, Sebastian, Christian Cherek, **Philipp Wacker**, Jan Borchers, and Simon Voelker (2020). "Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices". *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '20. Oldenburg, Germany: Association for Computing Machinery. ISBN: 9781450375160. DOI: 10.1145/3379503.3403538.

Lichtschlag, Leonhard, **Philipp Wacker**, Martina Ziefle, and Jan Borchers (2015). "The Presenter Experience of Canvas Presentations". Englisch (eng). *INTERACT 2015: Proceedings of the 15th IFIP TC.13 International Conference on Human-Computer Interaction*, pp. 289–297. ISBN: 978-3-319-22700-9. DOI: 10.1007/978-3-319-22701-6_21.

Wacker, **Philipp**, Kerstin Kreutz, Florian Heller, and Jan Borchers (2016a). "Maps and Location: Acceptance of Modern Interaction Techniques for Audio Guides". *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, pp. 1067–1071. ISBN: 9781450333627. DOI: 10.1145/2858036.2858189.

Wacker, **Philipp**, Oliver Nowak, Simon Voelker, and Jan Borchers (2019). "ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone". *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: ACM, 619:1–619:12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300849.

Wacker, **Philipp**, Oliver Nowak, Simon Voelker, and Jan Borchers (2020a). "Evaluating Menu Techniques for Handheld AR with a Smartphone & Mid-Air Pen". *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '20. Oldenburg, Germany: Association for Computing Machinery. ISBN: 9781450375160. DOI: 10.1145/3379503.3403548.

Wacker, **Philipp**, Adrian Wagner, Simon Voelker, and Jan Borchers (2018a). "Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality". *Proceedings of the Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: ACM, pp. 25–35. ISBN: 978-1-4503-5708-1. DOI: 10.1145/3267782.3267788.

Wacker, **Philipp**, Adrian Wagner, Simon Voelker, and Jan Borchers (2020b). "Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR". *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–11. ISBN: 9781450367080. DOI: 10.1145/3313831.3376848.

Book chapters (Edited, archival)

Arora, Rahul, Mayra D. Barrera Machuca, **Philipp Wacker**, Daniel Keefe, and Johann H. Israel (2021). "Introduction to 3D Sketching". *Interactive Sketch-Based Interfaces and Modelling for Design*. Ed. by Alexandra Bonnici, Kenneth P. Camilleri, Alfredo Ferreira, Johann H. Israel, Paul L. Rosin, and Metin Sezgin. (in print). River Publishers. Chap. 6.

Barrera Machuca, Mayra D., Rahul Arora, **Philipp Wacker**, Daniel Keefe, and Johann H. Israel (2021). "Interaction Devices and Techniques for 3D Sketching". *Interactive Sketch-Based Interfaces and Modelling for Design*. Ed. by Alexandra Bonnici, Kenneth P. Camilleri, Alfredo Ferreira, Johann H. Israel, Paul L. Rosin, and Metin Sezgin. (in print). River Publishers. Chap. 8.

Israel, Johann H., Mayra D. Barrera Machuca, Rahul Arora, **Philipp Wacker**, and Daniel Keefe (2021). "Input Processing and Geometric Representations for 3D Sketches". *Interactive Sketch-Based Interfaces and Modelling for Design*. Ed. by Alexandra Bonnici, Kenneth P. Camilleri, Alfredo Ferreira, Johann H. Israel, Paul L. Rosin, and Metin Sezgin. (in print). River Publishers. Chap. 7.

Wacker, **Philipp**, Rahul Arora, Mayra D. Barrera Machuca, Daniel Keefe, and Johann H. Israel (2021). "3D Sketching Application Scenarios". *Interactive Sketch-Based Interfaces and Modelling for Design*. Ed. by Alexandra Bonnici, Kenneth P. Camilleri, Alfredo Ferreira, Johann H. Israel, Paul L. Rosin, and Metin Sezgin. (in print). River Publishers. Chap. 9.

Posters (Peer-reviewed, non-archival)

Wacker, Philipp, Adrian Wagner, Simon Voelker, and Jan Borchers (2018b). "Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality". *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI EA '18. Montreal QC, Canada: Association for Computing Machinery, pp. 1–6. ISBN: 9781450356213. DOI: 10.1145/3170427.3188493.

Demos (Juried, non-archival)

Wacker, Philipp, Chat Wacharamanotham, Daniel Spelmezan, Jan Thar, David A. Sánchez, René Bohne, and Jan Borchers (2016b). "VibroVision: An On-Body Tactile Image Guide for the Blind". *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, pp. 3788–3791. ISBN: 9781450340823. DOI: 10.1145/2851581.2890254.

Theses

Wacker, Philipp (2012). "Twitter-Nutzung älterer Internetnutzer: Eine vergleichende Analyse". German (ger). Bachelor's Thesis. Aachen: RWTH Aachen University.

Wacker, Philipp (2014). "How Does It Feel? Presenter Experience and Evaluation While Using Canvas Presentation Tools". Englisch (eng). Master's Thesis. Aachen: RWTH Aachen University.

Fakultät für Mathematik, Informatik und Naturwissenschaften

RWTH AACHEN UNIVERSITY

Eidesstattliche Erklärung

Declaration of Authorship

I, Philipp Wacker

declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Hiermit erkläre ich an Eides statt / I do solemnely swear that:

1. This work was done wholly or mainly while in candidature for the doctoral degree at this faculty and university;

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated;

3. Where I have consulted the published work of others or myself, this is always clearly attributed;

4. Where I have quoted from the work of others or myself, the source is always given. This thesis is entirely my own work, with the exception of such quotations;

5. I have acknowledged all major sources of assistance;

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

7. Parts of this work have been published before as:

   Wacker, P.*, Wagner, A., Voelker, S., Borchers, J.: Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects in Augmented Reality. In Proc. ACM SUI '18.*

   Wacker, P.*, Nowak, O., Voelker, S., Borchers, J.: ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone. In Proc. ACM CHI '19*

   Wacker, P.*, Wagner, A., Voelker, S., Borchers, J.: Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR. In Proc. ACM CHI '20*

   Wacker, P.*, Nowak, O., Voelker, S., Borchers, J.:Evaluating Menu Techniques for Handheld AR with a Smartphone & Mid-Air Pen. In Proc. ACM MobileHCI '20*

Aachen, 31.05.2021 ______________________________

Index

Typeset May 31, 2021