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

RWTH AACHEN UNIVERSITY

Evaluating Guided Drawing Techniques in Handheld Augmented Reality

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

by
Martin Huppertz

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Torsten Wolfgang Kuhlen

Registration date: 29.06.2020
Submission date: 25.08.2020

Eidesstattliche Versicherung
Statutory Declaration in Lieu of an Oath

___________________________ ___________________________

Name, Vorname/Last Name, First Name — Matrikelnummer (freiwillige Angabe) Matriculation No. (optional)

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

I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

____

____

____

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

independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

___________________________ ___________________________

Ort, Datum/City, Date — Unterschrift/Signature

*Nichtzutreffendes bitte streichen

*Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt

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

Para. 156 StGB (German Criminal Code): False Statutory Declarations

Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.

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

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

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

Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence

(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.

(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

I have read and understood the above official notification:

___________________________ ___________________________

Ort, Datum/City, Date — Unterschrift/Signature

Contents

List of Figures

List of Tables

Abstract

When creating 3D models for the purpose of *Personal Fabrication*, the goal is usually to produce a physical object to place in a real environment. Since these objects often times need to fit other real geometry, the design process may involve the taking of many measurements. One approach to simplify this process is the ARPen, an *Augmented Reality* (AR) system, that allows for the creation of 3D models in-situ, through the use of a smartphone and a 3D printed pen.

Since many interactions with the ARPen rely on freehand 3D drawing, and since previous works have shown the limitations of freehand 3D drawing in *Mixed Reality* (MR), for this thesis we explored guided drawing techniques in AR that are based around drawing on virtual object surfaces. For this purpose we developed two drawing modes for the ARPen system: *closest point* and *raycasting*. The *closest point* technique places the drawing node on the nearest surface position on a virtual object, whereas the *raycasting* technique projects a ray from the ARPen's tip similar to a laser pointer, and places the drawing node at the ray's first intersection with a virtual object.

To evaluate these techniques, we conducted a user study. We investigated the quantitative impact that the techniques had on drawing precision while drawing on and around different geometry, compared to freehand drawing. Qualitative feedback from the participants was collected as well, ranking the drawing techniques in four categories. Overall, the *closest point* technique received not only the most positive feedback, but also had the highest average precision compared to the *raycasting* and freehand modes. While the *raycasting* technique received comparatively negative remarks from participants, it resulted in more precise results than the freehand mode, and could be improved by applying some changes such as providing more extensive visualization.

Überblick

Bei der Erstellung von 3D Modellen für Zwecke im Bereich *Personal Fabrication*, ist das Ziel für gewöhnlich, ein physisches Objekt zu produzieren, das in eine reale Umgebung platziert wird. Da diese Objekte oftmals mit anderen echten geometrischen Formen zusammenpassen müssen, kann der Gestaltungsprozess viele Messungen beinhalten. Ein Ansatz diesen Prozess zu vereinfachen ist der ARPen, ein *Augmented Reality* (AR) System, das die Erstellung von 3D Modellen "in situ" erlaubt, mittels der Benutzung eines Smartphones und eines speziellen 3D gedruckten Stifts.

Da viele Interaktionen mit dem ARPen auf freihändigem Zeichnen in 3D basieren und da bisherige Studien die Limitationen von freihändigem Zeichnen in 3D in *Mixed Reality* (MR) aufgezeigt haben, haben wir in dieser Arbeit geführte Zeichentechniken in AR untersucht, die für das Zeichnen auf virtuellen Objektoberflächen ausgelegt sind. Zu diesem Zweck haben wir zwei Zeichenmodi für das ARPen-System entwickelt: *closest point* und *raycasting*. Der *closest point* Modus platziert den Zeichenpunkt auf der nächsten Oberflächenposition eines virtuellen Objekts, während der *raycasting* Modus einen Strahl aus der Spitze des ARPen projiziert, ähnlich wie ein Laserpointer und den Zeichenpunkt an der ersten Überschneidung des Strahls mit einem virtuellen Objekt platziert.

Um diese Techniken zu bewerten, haben wir eine Benutzerstudie durchgeführt. Dabei haben wir den quantitativen Einfluss untersucht, den unsere Techniken auf die Präzision beim Zeichnen auf verschiedenen geometrischen Untergründen haben, verglichen mit freihändigem Zeichnen. Qualitatives Feedback der Teilnehmer wurde ebenfalls gesammelt, bei dem die Zeichentechniken in vier Kategorien bewertet wurden. Insgesamt hat die *closest point* Technik nicht nur das positivste Feedback erhalten, es hat außerdem die höchste durchschnittliche Zeichenpräzision geliefert, verglichen mit den *raycasting* und freihändigen Zeichenmodi. Obwohl die *raycasting* Technik vergleichsweise negatives Feedback von den Teilnehmern erhalten hat, hat es dennoch präzisere Resultate geliefert als freihändiges Zeichnen. Die Technik könnte verbessert werden indem Änderungen vorgenommen werden, zum Beispiel den Ausbau der visuellen Darstellung.

Acknowledgements

Firstly, I would like to thank Prof. Dr. Borchers and Prof. Dr. Kuhlen for examining this thesis.

I also want to thank my supervisor Krishna Subramanian for all his helpful and thoughtful advice.

Furthermore, I would like to thank everyone who participated in my user study.

Lastly, I want to thank my friends, my family, and especially Rachel, for supporting me throughout this thesis.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

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

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

Definition: *Excursus*

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

`myClass`

The whole thesis is written in Canadian English.

Download links are set off in coloured boxes.

File: myFile[a]

[a]http://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

Mobile handheld Augmented Reality has become widely available.

In recent years, both *Virtual Reality* (VR) and *Augmented Reality*(AR) have become widely accessible. VR has gained popularity through the release of consumer-grade VR systems such as the Oculus Rift[1] or the HTC Vive[2], whereas AR can now be implemented on smartphones due to frameworks such as ARKit[3] for iOs and ARCore[4] for Android. While high prices for *Head-Mounted Displays* (HMDs) remain a barrier to entry for VR, technological advancements in smartphones have made mobile handheld AR available to many people.

Creating models in the environment that they are meant to be placed in may be beneficial.

Since AR environments allow users to interact with virtual objects in a real environment, 3D modeling applications in AR are useful for the creation of 3D models that need to fit real world geometry. The ARPen[5] system (Wehnert [2018]) aims to provide an accessible way to create 3D models *in-situ*. It uses a 3D printed pen that is tracked by an iPhone running the ARPen application. The system provides the user an augmented view in which to create 3D models using both sketch-based interactions as well as classical pointer-based interactions.

[1]https://www.oculus.com/rift/ (Accessed: 04.08.2020)
[2]https://www.vive.com/ (Accessed: 04.08.2020)
[3]https://developer.apple.com/augmented-reality/arkit/ (Accessed: 04.08.2020)
[4]https://developers.google.com/ar (Accessed: 04.08.2020)
[5]https://hci.rwth-aachen.de/arpen

Freehand drawing is more difficult in 3D than in traditional 2D drawing.

Traditional drawing or sketching in 2D is usually done on a solid surface that guides every drawn stroke. When drawing in 3D however, this kind of guidance is usually not available. This leads to some common problems with freehand 3D drawing in both VR as well as AR. Novice users align their strokes so that they look correct from their view, but are not positioned correctly in 3D. Even when users consciously try to align strokes in three dimensions this is not an easy task and is a source of errors (Arora et al. [2017]). Experiments by Wacker et al. [2018] in AR have shown that freehand drawing precision can be improved by both physical guidance as well as visual guidance. Moreover, VR and AR drawing applications have tried to constrict the user's drawn strokes to 2D planes in 3D space, in order to recreate a pseudo-2D drawing interaction (Barrera Machuca et al. [2017], Xin et al. [2008]).

In this work, we developed drawing techniques that are based around drawing on object surfaces.

With this in mind, in this thesis we aimed to create drawing techniques for the ARPen that allow the user to draw on the surfaces of different kinds of virtual geometry. The goal for these drawing techniques was to improve drawing precision while remaining intuitive in use. We then evaluated these drawing techniques in a user study.

We tested the drawing techniques in a user study.

We will begin this thesis with an overview of the domain of AR, and an introduction to the ARPen system (Chapter 2 "Background"). Afterwards, we will present related work, focusing on sketch-based modeling and research in freehand sketching performance in AR and VR (Chapter 3 "Related Work"). In the following Chapter 4 "Design and Implementation", we will describe the design of our drawing techniques, including iterative changes and discarded features, as well as the implementation of the drawing techniques. In Chapter 5 "User Study", we will describe the setup and procedure of the user study we conducted to evaluate the drawing techniques. After that, in Chapter 6 "Results" we will present and analyze the results obtained in the aforementioned user study. We will conclude with Chapter 7 "Summary and Future Work", summarizing the results of this thesis and outlining possible future work.

Chapter 2

Background

To give context for this thesis, in this chapter we will give an introduction to *Augmented Reality* (AR) as a concept, before discussing different technical aspects of implementing AR, including display types, tracking and interaction methods, as well as perception issues in AR. We will conclude with an overview of the ARPen system, which the work in this thesis is based on.

2.1 Augmented Reality

In Milgram's virtuality continuum, *Augmented Reality* (AR) is defined as a mostly real environment, which is "augmented" by having virtual objects superimposed onto or integrated into the scene. In this chapter we will give an overview of AR displays, ways to register and track real-world objects and environments, common interaction methods in AR, as well as perception issues in AR (Azuma [1997], Azuma et al. [2001], Milgram and Kishino [1994]).

2.1.1 Displays

There are a variety of display types for AR systems. The most common types include *Head-Mounted Displays*

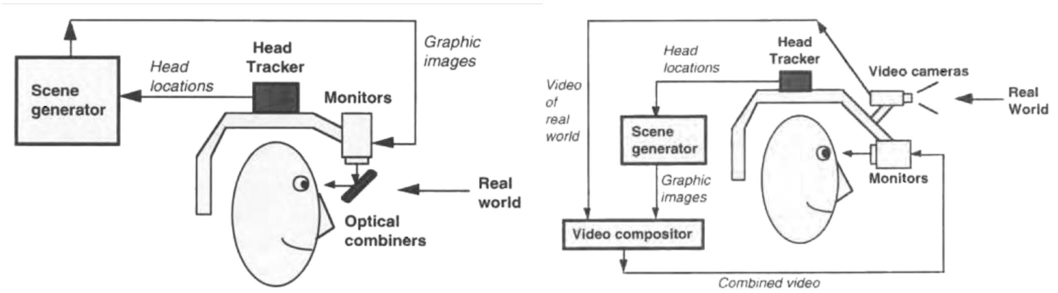

Figure 2.1: Left: Conceptual diagram of an optical see-through HMD. Right: Conceptual diagram of a video see-through HMD. Image taken from [Azuma, 1997].

Common types of AR displays can be categorized into head-mounted displays, handheld displays, and projection displays.

(HMDs), *handheld displays* and *projection displays*. HMDs can be categorized into *optical see-through* displays and *video see-through* displays. With *optical see-through* displays the user looks at the real world through a semi-transparent pane onto which virtual content is projected. In *video see-through* displays the user observes the world through a screen on which a camera feed is displayed. *Projection displays* include systems in which one or more projectors are used to project virtual content onto the environment. The images need to be stretched or distorted in a way so that they look geometrically correct from the viewpoint of the user (Azuma [1997]). A common form of handheld AR displays today includes smartphones, where the display shows an AR-altered image from the front facing camera, serving as a "magic lens" to an augmented view of reality. For the remainder of this chapter we will focus on handheld AR displays in particular.

2.1.2 Registration

Smartphones can track their own position in real world environments.

One of the issues that any AR system needs to address is the problem of *registration*, that is, a way to track the real world and real objects. To simulate a functioning AR environment, virtual and real-world objects need to be aligned properly. To align virtual and real-world objects, a coordinate system is needed which matches the real world and in which virtual objects can be placed. Modern AR frame-

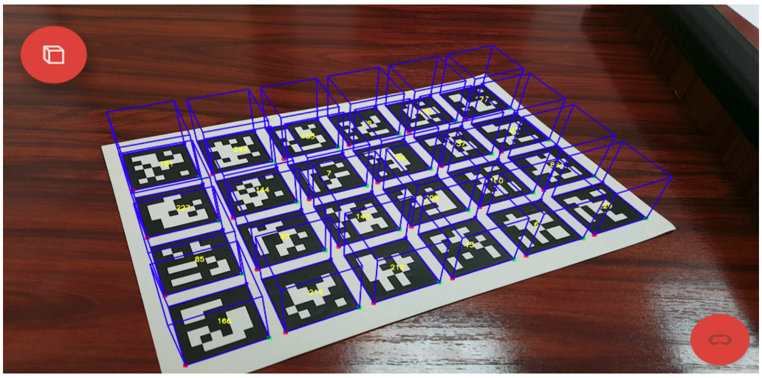

Figure 2.2: Example of *ArUco* fiducial markers.

works such as Apple's *ARKit*[1] or Google's *ARCore*[2] provide sophisticated tracking methods which use the smartphone's camera image, gyroscope, accelerometer, and other sensors to create a stable and coherent AR session. Another method of registering the real world is using fiducial markers. These markers are commonly black and white shapes, which encode some information that is commonly used to identify the real-world objects that display these markers. One framework for the generation of such markers is *ArUco*[3].They can be used to provide references for the AR tracking, by fixing them onto static objects in the environment, or they can be used to mark mobile objects, so that their movement can be tracked accurately. (Azuma [1997]).

Fiducial markers can be used to track specific objects.

2.1.3 Interaction Methods

AR applications use a wide array of interaction methods. For handheld devices the interaction methods can be categorized into *embodied interaction* and *tangible interaction*. In an *embodied interaction*, the interaction occurs via the device itself, such as device motion or the touchscreen. One way to interact with virtual objects through the touchscreen is

Interaction can occur with the handheld device itself, through gestures, or through external tangible objects.

[1] https://developer.apple.com/augmented-reality/arkit/ (Accessed: 04.08.2020)

[2] https://developers.google.com/ar (Accessed: 04.08.2020)

[3] https://www.uco.es/investiga/grupos/ava/node/26 (Accessed: 04.08.2020)

ray picking. Here, the user picks an object, by tapping on the screen and a virtual ray is projected through the virtual environment at the corresponding screen coordinates. The first object hit by this ray is then selected. *Tangible interaction* presents a more direct way of manipulating virtual objects. Here the physical manipulation of real world objects is interpreted as input. The object's position, orientation, and presence or absence in the view are common parameters used to trigger certain actions (Gervautz and Schmalstieg [2012]).

2.1.4 Depth Perception Issues in AR

Insufficient depth cues can cause users to have trouble interpreting visual content.

Depth perception is important in AR, since issues with depth perception may cause the user to have trouble in interpreting the content he sees. This may take the form of ambiguous depth ordering or unclear object relationships, among other things. Most of these issues arise because the human visual system relies on certain depth cues that are often times missing in AR scenes. Some of these issues lie in the rendering of the augmented scene: missing shadows, incorrect occlusion (such as when a distant virtual object occludes a nearby real object), and other *pictorial depth* cues, may cause the viewer to draw false conclusions about the positions and sizes of objects, or their positions and sizes may be ambiguous.

A lack of stereoscopy is a common issue in handheld AR.

The display device can be another problem source. Consider handheld displays. Many handheld displays do not have the capability for *stereoscopic* vision, which eliminates several depth cues, such as *binocular disparity*, which is the depth information gained by having two views of a known horizontal offset, and *physiological depth cues*, such as *vergence* and *accommodation*.

Definition: *vergence*

VERGENCE:
The rotation of the eyes towards each other, to focus on a point at a certain depth.

ACCOMMODATION:
Flexion of the eye's lens, to focus on a point at a certain depth.

Definition: *accommodation*

Handheld displays shift the user's viewing angle.

Moreover handheld displays rely on viewing the augmented scene through a camera view displayed on a screen that is not aligned with the user's view, in contrast to HMDs, where the cameras are set up to represent the human eyes as well as possible. This creates a *viewing angle offset* that changes with the motion of the display. Even if the user aligns the display in the middle of their view, the camera's *Field of View* (FOV) will cause a distorted view.

A plethora of other issues may arise from insufficient technical quality. Insufficient image resolution, framerate, exposure, contrast, and color correctness may impact the users perception. Errors in the registration may cause virtual and real objects to become misaligned (Kruijff et al. [2010]).

These perception issues affect mid-air drawing performance, which will be discussed more in Section 3.2.

2.2 The ARPen System

The ARPen is a handheld, AR based 3D modeling system developed at the Media Computing Group of the RWTH Aachen. It consists of a 3D printed pen that is tracked by an iPhone running the ARPen iOS application. The software uses Apple's *SceneKit*[4] framework for 3D rendering purposes and uses *ARKit*[5] in order to provide the basic AR functionality. *ARKit* provides natural feature tracking, anchoring the virtual scene to the real world. It uses the iPhone's gyroscope and camera feed to estimate the phone's real-world orientation and position.

[4] https://developer.apple.com/documentation/scenekit/ (Accessed: 04.08.2020)

[5] https://developer.apple.com/augmented-reality/arkit/ (Accessed: 04.08.2020)

Figure 2.3: The ARPen system in use. The marker box at the end of the pen needs to be in the camera frame to be tracked correctly.

The ARPen is tracked by fiducial markers, and has buttons whose signals are transmitted via bluetooth.

The pen itself has a box at its end with six fiducial markers attached to the sides. The framework used to generate and track these markers is *ArUco*[6]. The markers consist of a square pattern of black and white squares, which encode a numeric ID. The patterns are non-symmetrical, which allows them to be recognized at any rotation and most angles. Since the patterns are high in contrast, they are easy to track in most lighting conditions. The pen is also equipped with three buttons and a Bluetooth chip, allowing the application to access the state of the buttons for its interaction.

Previous research on the ARPen has investigated interaction methods for object rotation ([Klamma, 2019]), advanced 3D modeling techniques ([Benscheid, 2019]), menu techniques ([Wacker et al., 2020a]), mid-air pen position visualizations ([Wacker et al., 2020b]), and different physical pen designs ([Schäfer, 2020]).

[6]https://www.uco.es/investiga/grupos/ava/node/26 (Accessed: 04.08.2020)

Chapter 3

Related work

Since a primary application of the ARPen is in-situ 3D modeling, we will take a look at research related to sketch-based 3D modeling techniques. We will start off by presenting several sketch-based modeling applications and discussing their merits, before going on to presenting research on sketching performance in VR and AR. A detailed overview of the related work is presented in table 3.1

3.1 Sketch-based 3D Modeling

In this section we will present related works where sketches are used to generate 3D models. There are different approaches to generating 3D models through sketch-based interaction. One approach is to automatically generate 3D models from finished traditional 2D sketches. This is an interesting field of research, since traditional 2D sketching is one of the fundamental steps in visual design. However, an inherent problem to this is the fact that a single 2D sketch contains insufficient information to generate a full 3D model.

Creating 3D models from finished sketches is difficult.

Another approach is to create an interface that uses sketch-based interactions for 3D modeling. "ILoveSketch" by Bae et al. [2008] is a system that stays close to the sketching in-

Category	Source	Features
2D sketch-based interactions	ILoveSketch [Bae et al., 2008]	- Sticks close to sketching metaphors Does not create solid 3D models, only 3D sketches
	Teddy [Igarashi et al., 2006]	- Add features to basic shapes through sketch-based operations
	FiberMesh [Nealen et al., 2007]	- Similar to Teddy, but adds ability to edit existing shapes
3D sketching evaluations	3D sketching Performance [Arora et al., 2017]	- Planar alignment is a problem in 3D freehand sketching
	Physical Guides [Wacker et al., 2018]	- Drawing around physical or virtual guide objects improves drawing precision
Guided 3D sketching implementations	Multiplanes [Barrera Machuca et al., 2017]	- Application suggests drawing planes automatically
	Napkin Sketch [Xin et al., 2008]	- Projects drawing onto user-defined planes

Table 3.1: Summary of the related work presented in this chapter.

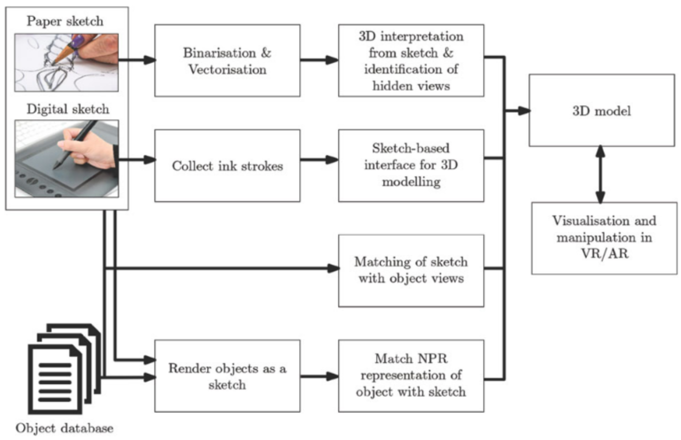

Figure 3.1: Overview of sketch-based 3D modeling approaches. Image taken from [Bonnici et al., 2019]

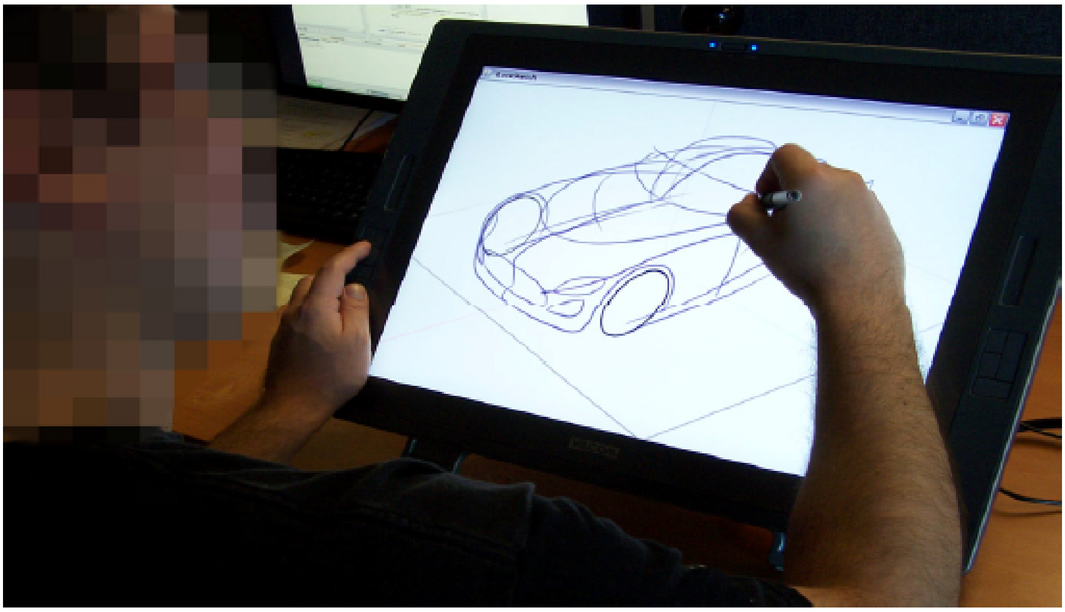

Figure 3.2: The interface of "ILoveSketch" ([Bae et al., 2008]).

teraction. Unlike other in applications, that attempt to create solid 3D models consisting of volumes, the user can create sketches consisting of individual strokes in 3D space.

Often times the sketch-based interface is focused on adding details to basic shapes in order to create more sophisticated geometry. One example of this is "Teddy" (Igarashi et al. [2006]), a system that allows for the creation of 3D objects by drawing a silhouette, and adding features by drawing an outline on the existing object and extruding geometry out from there . A similar system is "FiberMesh", developed by Nealen et al. [2007] that takes the same approach as Teddy, but adds the ability to refine drawn shapes. The geometry of an object is defined by its "control curves". These are drawn onto an object and can thereafter be manipulated by dragging them. They can be hard or soft edges. Since these curves can be manipulated after creation this allows the user to edit existing models, in addition to the ability to create new ones from scratch. This interaction is similar to traditional digital 3D modeling, where models can be edited by moving existing vertices, edges or faces.

Sketch-based interaction can be used to add details to existing models.

FiberMesh allows the user to define detailed object by adding sharp or soft edges.

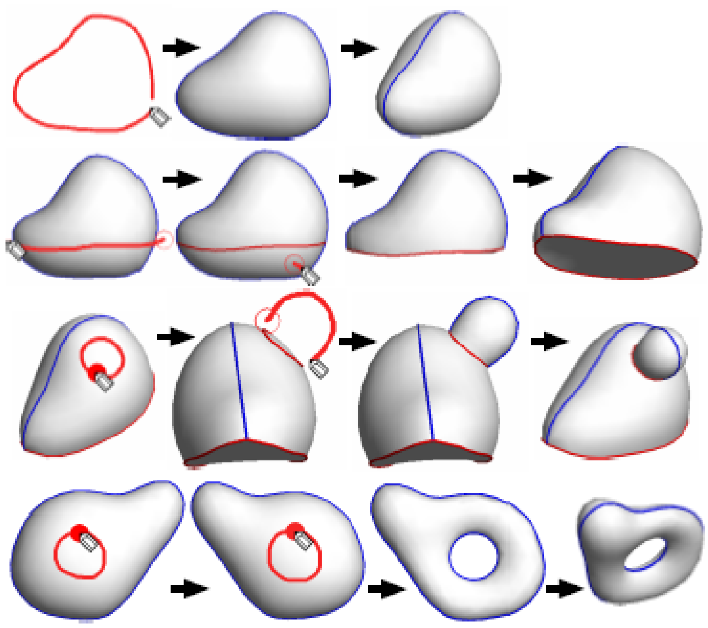

Figure 3.3: Example workflow in "FiberMesh" ([Nealen et al., 2007]).

3.2 Sketching in Mixed Reality

The previous examples have dealt with sketch-based interactions on 2D touchscreen interfaces. In this section we will present related works that investigate 3D drawing performance in VR and AR, as well as sketch-based modeling applications that rely on three dimensional interaction.

3.2.1 Freehand 3D Sketching

Aligning strokes with planes is difficult in 3D freehand sketching.

Experiments by Arora et al. [2017] showed that 3D sketching creates issues that are not present in traditional 2D sketching. In traditional sketching the presence of a physical drawing surface constrains the interaction to planar curves. In freehand 3D sketching however, if no such guidance is given, the user can draw non-planar curves, which means that strokes that appear to be drawn correctly from

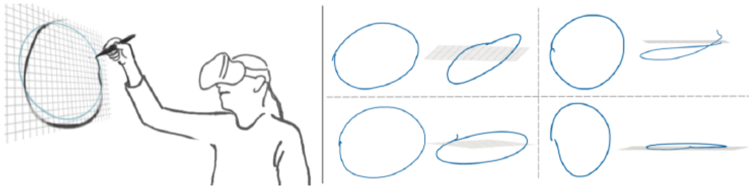

Figure 3.4: Drawing in VR creates difficulties with the planar alignment of strokes. Image taken from [Arora et al., 2017].

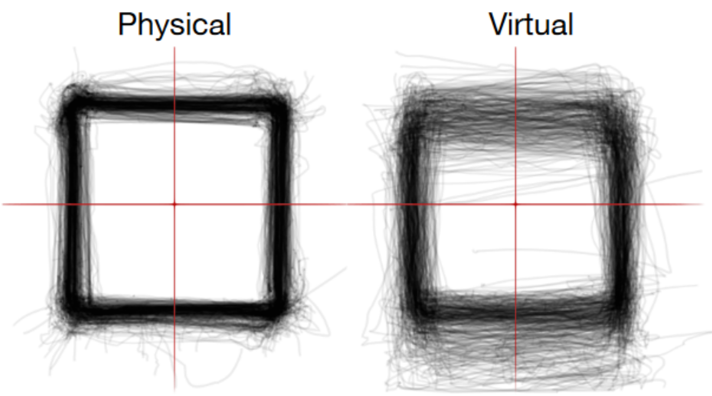

Figure 3.5: Experiments by Wacker et al. [2018] showed that while both physical as well as virtual guides improve drawing precision, physical guides are more effective.

the user's perspective, may be misaligned with the plane they intended to draw on (Figure 3.4). Designers participating in the study recommended projecting strokes onto desired virtual planes for greater accuracy.

Experiments by Wacker et al. [2018] investigated the impact of both physical and virtual guides on 3D drawing performance in AR. Physical guides were shown to have the most positive impact on drawing performance, although virtual guides improved the drawing performance as well.

Guidance objects to draw around improve drawing precision.

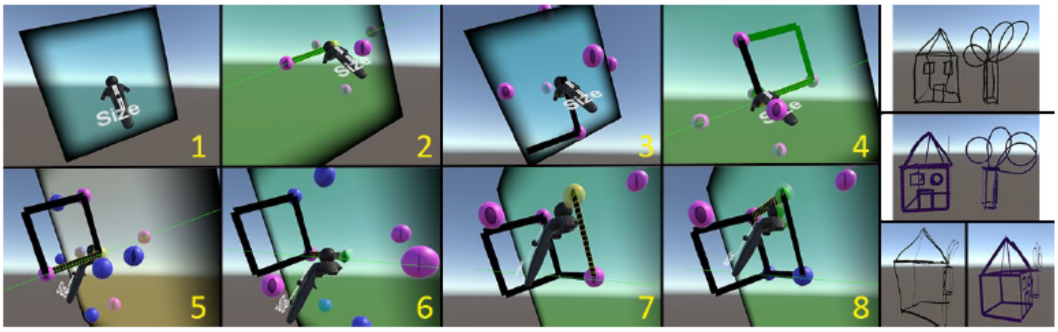

Figure 3.6: Drawing process in ”Multiplanes” ([Barrera Machuca et al., 2017]).

3.2.2 Assisted 3D Sketching

Since the ARPen is designed as an in-situ AR authoring device, it has to be assumed that users do not have access to adequate physical guides while drawing. Thus, a virtual guidance system may be desirable.

In Multiplanes the system suggests guidance planes automatically.

Barrera Machuca et al. [2017] created ”Multiplanes”, a VR drawing system that allows for assisted freehand drawing. In Multiplanes, beautification trigger points (BTPs) appear as nodes in midair while the user draws. They suggest possible geometric shapes that the user might be trying to draw, and if the drawn stroke starts or ends in a BTP, the line is snapped to the corresponding geometry and beautified. Moreover guidance planes are generated automatically, onto which drawn strokes are projected .

In Napkin Sketch the user can move the tablet or the AR reference image to change his viewing angle.

”Napkin Sketch” (Xin et al. [2008]) provides assistance to the user while sketching as well. Napkin Sketch consists of a tablet PC with a touchscreen and front-facing camera, and a ”napkin” consisting of a sheet of paper with fiducial markers printed on it, which serves as an anchor for the virtual content. In Napkin Sketch, the user can define 3D planes onto which drawn strokes are projected. This technique works similar to ray picking (see Section 2.1.3), in that the 3D location of the drawn stroke is determined by casting a ray from the camera through the screen space that the pen is drawing on. The user can move the screen around the napkin, or manipulate the napkin itself so that they have a comfortable angle from which to draw.

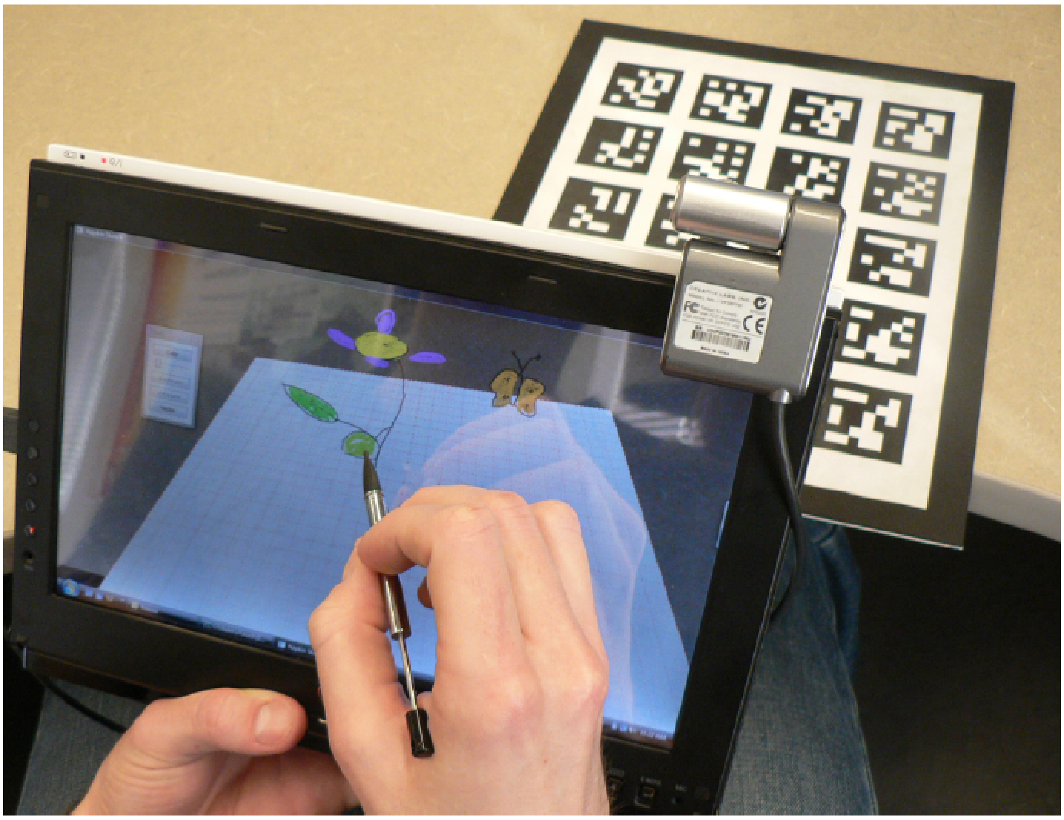

Figure 3.7: The interface of "Napkin Sketch" ([Xin et al., 2008]).

Chapter 4

Design and Implementation

4.1 Design

The current ARPen app supports freehand drawing, however past user studies showed that users have trouble with this in 3D space. For this thesis, we wanted to explore drawing modes that rely on drawing on the surfaces of virtual objects instead of free 3D drawing. We designed two guided drawing modes. Both modes require a virtual object to be present on which the user can draw.

Guided drawing techniques are designed to draw on object surfaces.

We called the first mode *raycasting*. In this mode, a virtual ray is projected from the ARpen's tip similar to a laser pointer. It detects the first virtual object in its path and positions a `SCNNode`, called `projectionNode` to that hit location and makes it the active point at which drawn lines are placed. The second mode, called *closest point* mode, positions the `projectionNode` on the closest surface point on any virtual object in the scene.

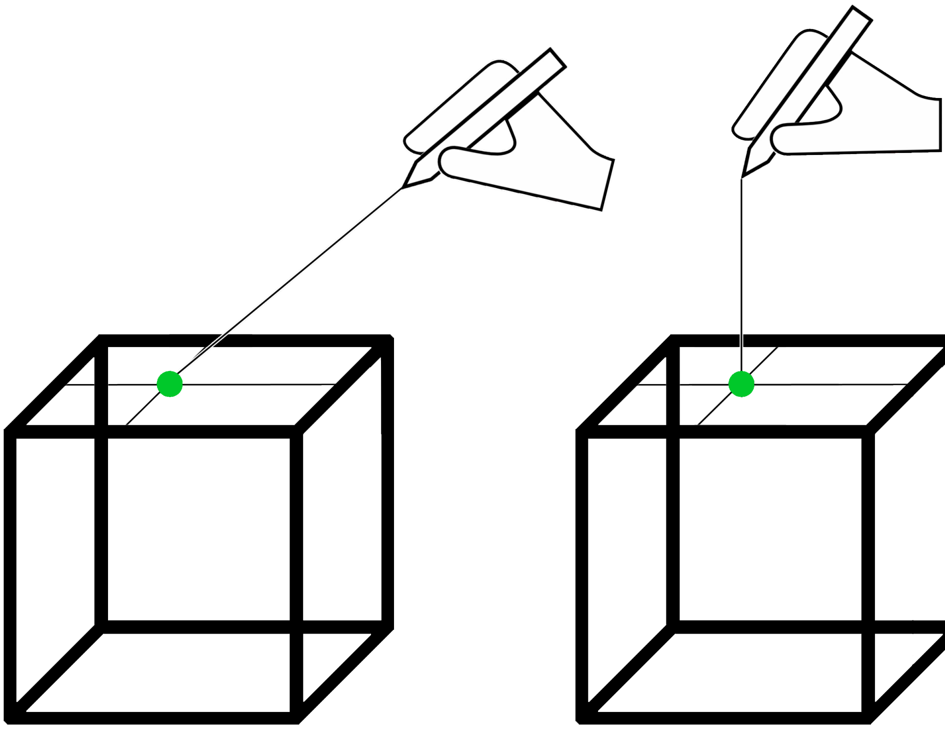

Figure 4.1: Left: Concept of the *raycasting* technique. Right: Concept of the *closest point* technique.

4.2 Implementation

The guided drawing modes are implemented as plugins, similar to the existing freehand `PaintPlugin` and object creation plugins. The default freehand mode uses the `drawingNode` as the point at which lines are drawn and objects are placed. The guided drawing modes use the `projectionNode` as a drawing point, since it is removed from the pen tip and since creating a separate node for the guided drawing plugins caused less issues within other plugins than using the `drawingNode`.

4.2.1 Raycasting Guide

Initially the orientation of the pen was not accessible.

To implement the *raycasting* guide, we first needed to project a ray from the pen tip along its length. The previous implementation of the ARPen app detects the ArUco markers on the physical pen and uses them to calculate the pen tip's position. However, the orientation of the pen is not calculated or stored in the `pencilPoint` node. We solved this with a quick workaround: the function that calculates the position of the pen tip can create a pen tip node depending on the pen's length. We used this function to create a second pen tip at twice the pen's length and called it `directionNode`. Using this node, we know the direction in which to project a ray.

Once we have a second position to direct the ray from the pen tip, we use SceneKit's function `hitTestWithSegment` to check the segment of space in front of the pen tip for any virtual object that might be a suitable drawing surface. Since the function detects any `SCNNode` with a geometry property, some nodes needed to be excluded from the search, like the pen tip and drawing nodes themselves. If a valid object is hit by the ray, the `projectionNode` is positioned there, and drawing is enabled. If no valid object is available, the drawing button is disabled.

4.2.2 Closest Point Guide

The closest point function calculates the surface position on an object that is closest to the pen tip and the distance that position has to the pen tip for each valid object in the scene. It then selects from them the position with the closest distance to the pen tip. The `projectionNode` is then positioned at the selected surface location.

4.3 Iterations

4.3.1 Raycasting Guide

One major design issue with the *raycasting* guide is how it behaves when the user does not hit an object with the targeting ray. The initial implementation of the *raycasting* guide allowed the user to draw, even if the targeting ray does not hit any object. This led to undesirable behavior and so we changed it to only accept the drawing input if a valid target point is available.

It can be difficult for the user to differentiate whether the ray does not hit any object or whether the marker is not detected.

In this changed implementation, the projected drawing point sticks to the last valid location that the ray hit. This created another issue in that the projected drawing point would not move if the ray missed the object. This looks a lot like when the ArUco markers of the ARPen move outside the camera frame and the tracking is lost, which created some confusion for initial testers. We tried disabling the projected drawing point if the ray was not hitting an object, however this caused more confusion than before. For this reason we decided to display the regular pen tip node in a semi-transparent way, to show that the tracking was still working correctly.

4.3.2 Closest Point Guide

We optimized the amount of objects the function considers.

For the implementation of the *closest point* guide, we first tried collecting all objects in the scene and then filtering them down to just the relevant guide shapes. This proved to be unnecessarily difficult, since all the objects that are considered valid guide shapes are created in the plugin functions and are all stored as children of the pen's `drawingNode` and all have specific names. So, we search the `drawingNode`'s child nodes by name instead to get valid objects as input for the *closest point* function.

Another issue with the *closest point* mode is the drawing behavior on interior corners or between objects. In those cases there are often two valid nearest points available, and the

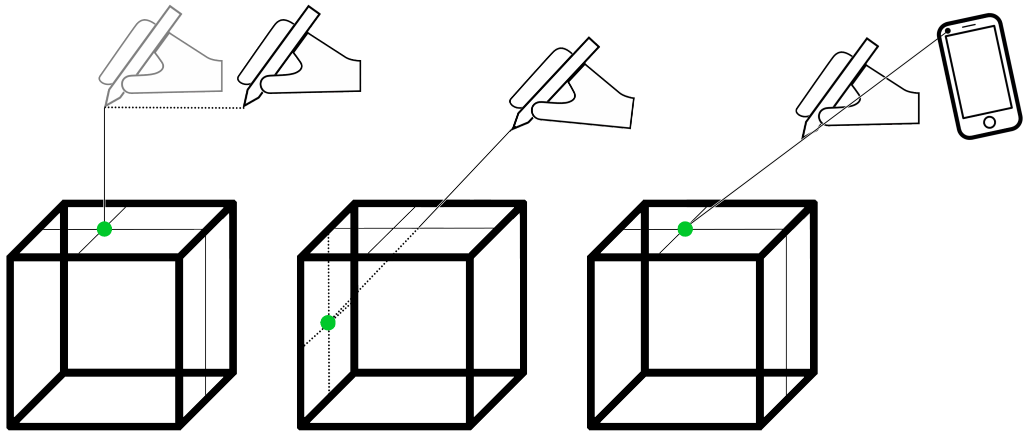

Figure 4.2: Left: Concept of the "closest point with offset" technique. Middle: Concept of the "click-through raycasting" technique. Right: Concept of the "raycasting from camera" technique. These techniques were considered but not implemented.

drawing mode switches between them frequently, leading to jittering behavior.

4.4 Other drawing modes

When deciding which drawing modes to implement, there were some designs we discarded, but which might be interesting nonetheless. The first is the "click-through" mode, an add-on to the *raycasting* mode. This technique works like the *raycasting* technique, except that the user may choose for the ray to ignore the first n intersections, which would allow the user to draw on the far side of an object without having to maneuver around it. The second is an "offset" mode for the *closest point* mode. The *closest point* mode allows the user to draw accurately and comfortably with small movements from far away, as long as the user stays on the side of the object they are drawing on. If, however they want to draw around a corner, they have to maneuver around to that corner. With the option to select an "offset", the user would have the option to do that same maneuver with less motion. Since these two techniques are niche add-ons to the existing drawing modes that require some

There were some drawing modes we considered, but decided not to implement.

experience with the base techniques, we decided to stick to the more basic versions of the *closest point* and *raycasting* modes.

Raycasting from the camera may be an intuitive interaction method similar to a mouse pointer or touch pen.

Another technique we considered is raycasting from the camera view towards the pen tip. This technique is interesting since it caters to the intuitive behavior observed in many untrained ARPen users: They maneuver the pen so that the pen tip in the augmented screen view is over the position that they want to draw on. This mostly leads to incorrect depth positions. Casting a ray through the pen along the camera view would correct that depth error and cause the user to draw on the corresponding surface position. This technique was rejected since its interaction is very similar to that of Napkin Sketch (Xin et al. [2008]), and in a way it reduces the ARPen to a 2D cursor, while the other drawing modes rely on three dimensional interactions. Moreover, in this mode users have to maneuver in such a way that they have a direct view from the camera to the point they want to draw on, while in the other drawing modes they can draw behind obstructing geometry.

Chapter 5

User Study

We performed a user study in order to evaluate our implemented guided drawing techniques. We gathered quantitative data in order to estimate the impact that the drawing guides have on user performance, and to identify use cases in which one drawing mode may be more useful than another. Additionally, we collected feedback using a questionnaire to evaluate the users' preferences, understand what factors may have affected the users' performance, and identify possible improvements to the drawing guides.

Since we wanted all users to use all drawing modes in order to compare them, we used a within-group design for the study. Using the two guided drawing modes *closest point* and *raycasting* as well as the default *freehand* mode, participants were tasked with tracing three-dimensional lines on the surfaces of virtual 3D objects. These tasks will be described in more detail in Section 5.1.

All participants tested all drawing techniques.

First the users were given a brief introduction to the ARPen app, and had the opportunity to familiarize themselves with drawing in AR. This was done to eliminate known drawing errors, previously observed in ARPen users, such as drawing in a 2D plane, ignoring the third "depth" dimension. Afterwards, the users were asked to trace target lines on 10 virtual cubes for each drawing mode. To counteract learning effects the order of the drawing modes was determined using a Latin square. The order of the cubes was randomized.

At the beginning of the study, the participants received an introduction to using the ARPen.

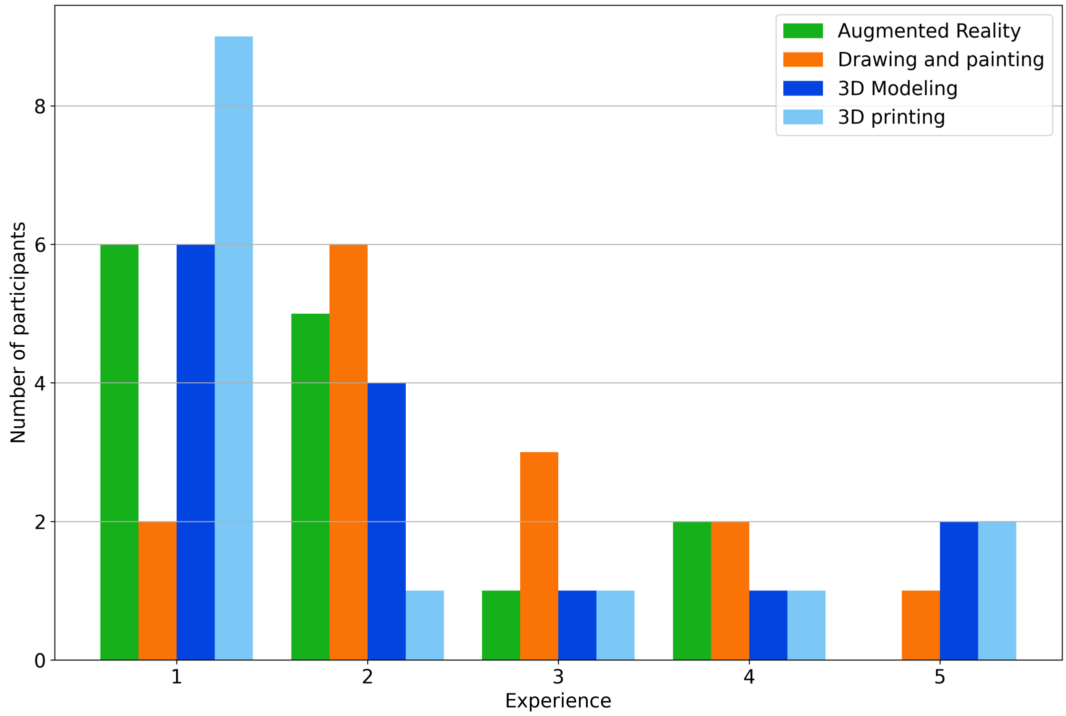

Figure 5.1: Results of the questionnaire regarding prior experience. Participants were asked to rate their experience in related fields on a scale of 0 to 5, with 0 being ”no experience”, and 5 being ”much experience”. The majority of participants had little to no experience with AR, 3D modeling or 3D printing.

The participants performed drawing tasks while seated at a table.

For the duration of the study the participants were seated at a table. Although they were asked to remain seated while drawing, they were encouraged to lean to the sides or the front to change their viewpoint. They were also allowed to stand and move around the virtual object to examine the results of their drawing after a completed task. We asked the participants to hold the ARPen in their right hand and the iPhone in their left hand, in any grip they prefer. They were asked not to rest their arms or elbows on the table while drawing. The participants were encouraged to think aloud while completing the presented tasks. A common problem with the ARPen was that often times the app would lose track of the physical world, causing all virtual objects to drift off from their assigned positions. To resolve this issue, a reference image was fixed to the table which the ARKit could use as an anchor point.

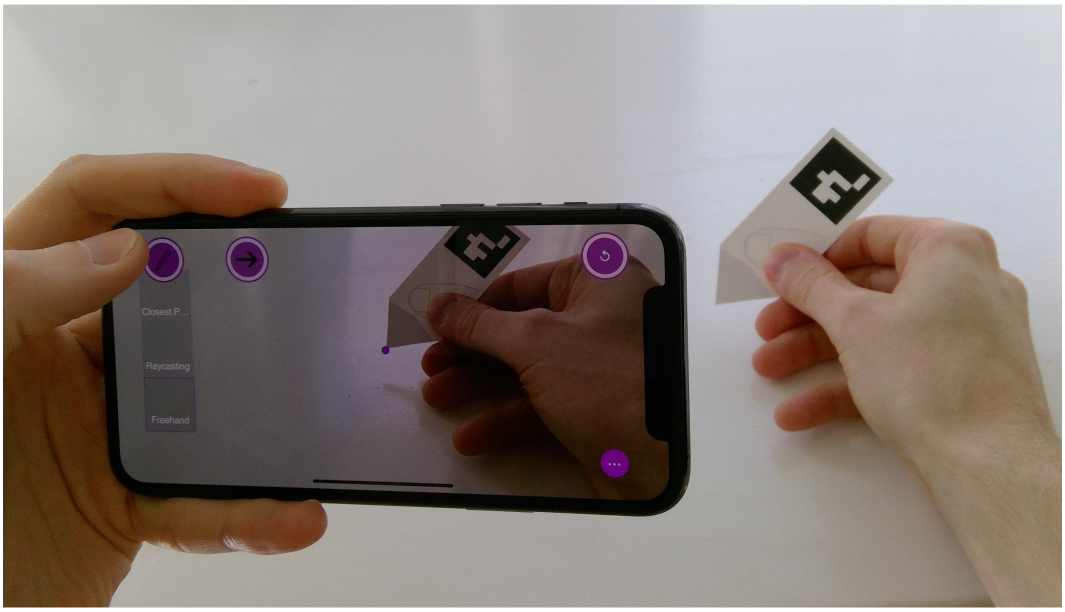

Figure 5.2: The cardboard version of the ARPen used in the study.

We recruited 14 participants for the study, including 8 males and 5 females, with ages ranging from 20 to 33 years. All participants were right-handed and able-bodied.

Hygienic safety measures were taken for the study.

Since the study was conducted during the start of the 2020 coronavirus pandemic, special hygienic measures were taken. The study was conducted outdoors to improve ventilation. This had no impact on visibility, however the low temperature caused some users to have shaky hands after around 45 minutes, resulting in some incomplete trials. Since we used a Latin square design for the order of the drawing techniques, and randomized the order of the individual tasks, all conditions were affected similarly. Moreover the users wore masks and latex gloves. These measures did not seem to affect user performance at all. Also, instead of using the 3D printed version of the ARPen, a disposable cardboard printout version was used. The cardboard version has three key differences compared to the 3D printed version: First, it is a passive pen without a bluetooth chip, meaning that software buttons on the smartphone's touchscreen were used instead of the buttons on the ARPen. Second, it has only one fiducial marker. And third, it is shorter in length. While the first two points are

A cardboard version of the ARPen was used for the study.

limitations of the cardboard version, the shorter length is a benefit. Since the pen tip and the marker are closer together, it is easier for the user to keep the marker in the camera view. The fact that it only has one marker limits the range of orientations that the pen can be used in, which affects the *raycasting* mode more than the other modes.

5.1 Task Descriptions

There were three independent variables: *drawing mode*, *size*, and *CornerType*.

The cubes used in the different tasks have two variable properties: Size and the location of the target line. The small cubes have an edge length of 8 cm, so that the participants would need more precision to maneuver the smaller geometry. The large cubes have an edge length of 12 cm, so that the participants would have to use their whole range of motion to maneuver between the front and back facing sides of the cube. We called this variable *CornerType*. The target line had five different variations: middle, interior corner, top, right, and front. In the middle condition, the target line goes around the four vertical sides of the cube, in the interior corner condition, the cube is bisected by a plane, creating a concave edge at the target line. In the conditions top, right, and front the target line is at the edges of one of the cube's sides.

The target line variations were chosen investigate whether the underlying geometry affects the drawing performance, in particular whether the techniques perform better or worse on concave or convex geometry. We were also interested in evaluating whether the orientation of the target line would affect drawing performance.

This results in the number of trials: 3 (drawing mode) x 2 (size) x 5 (CornerType) = 30 trials per participant.

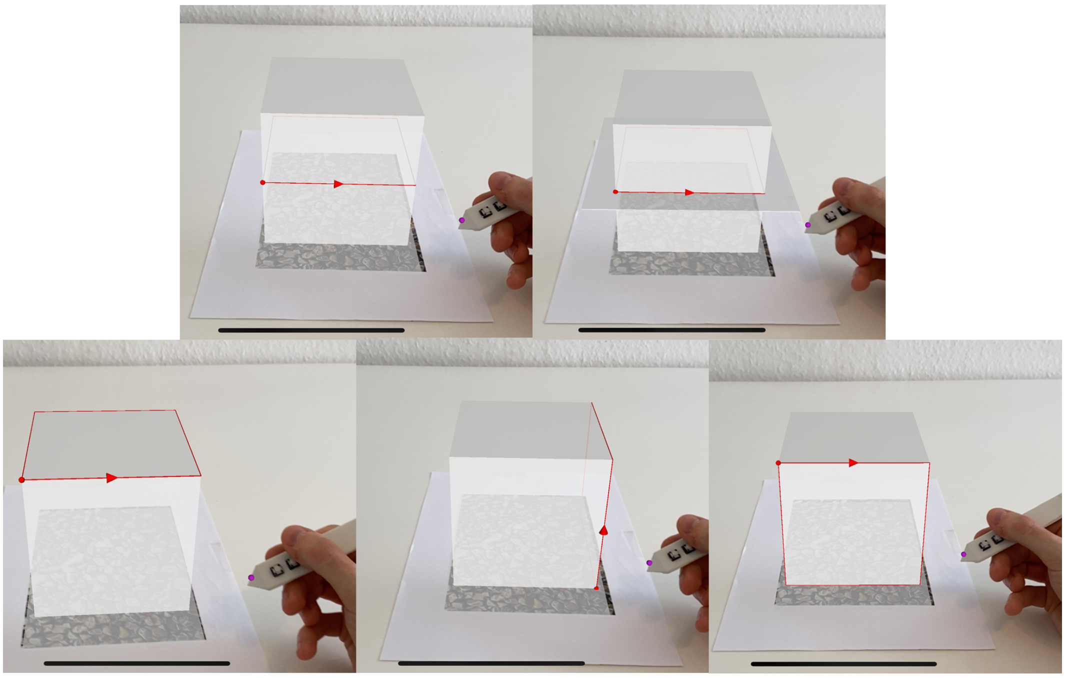

Figure 5.3: Displayed are the five conditions of the *CornerType* variable: from top left to right the conditions are *middle* and *interior*, from bottom left to right the conditions are *up*, *right*, and *front*.

Chapter 6

Results

In the following sections we will first present the qualitative results of the questionnaire and the participants' comments throughout the study. After that we will present the quantitative data we collected from the participants' performance in the study.

6.1 Qualitative Results

In this section we present the qualitative results of the questionnaire, our observation of the users' performance and their comments during the study.

Participants ranked the drawing modes in four categories.

The questionnaire included questions about the participants' demographic, followed by questions about their experience with the ARPen in general, before asking the participants to rank the drawing modes they used during the study in terms of personal preference, intuitiveness, perceived precision, and ergonomic comfort. It also included text fields to give more detailed feedback about the drawing modes. Participants also used these text fields to give general feedback about the ARPen.

In general, users reported having difficulties thinking in 3D and using the real world as reference in the beginning but

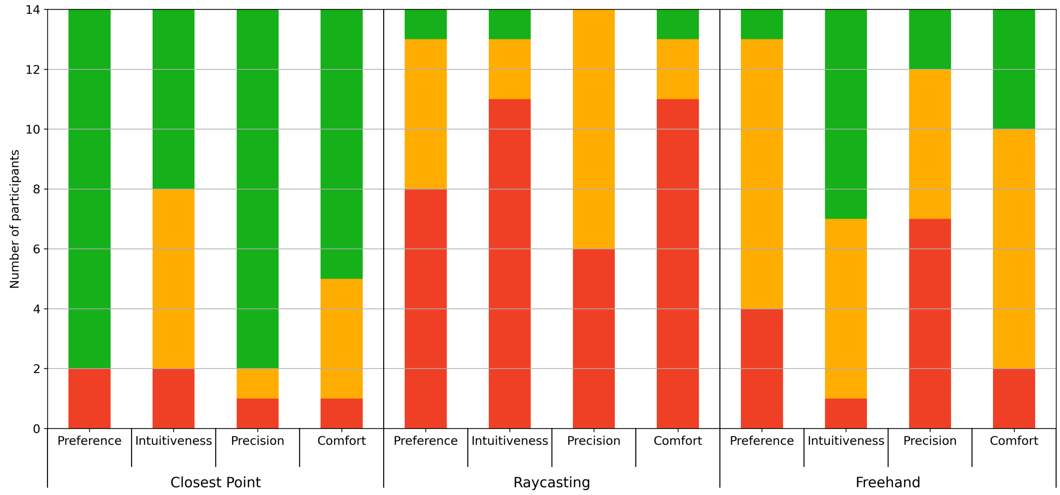

Figure 6.1: Rankings of the drawing modes in different categories. The ranking "Best" is depicted in green, "Middle" is orange, and "Worst" is red. The *closest point* technique received mostly positive feedback across all categories, while the *raycasting* technique received mostly negative feedback.

adapting to this over time. Some users also commented that using a head mounted AR display would be preferable to holding a smartphone. Users also stated that it was often difficult to estimate the pen tip's position in 3D space, especially when working on the back-facing side of the virtual object. Some went on to suggest adding more depth cues, such as orthogonal rays or planes projecting from the pen tip. The tracking of the pen received overall positive comments, although some users had issues keeping the AR markers of the pen in the camera frame.

Participants had trouble judging the depth positions of virtual objects.

Most participants preferred the *closest point* technique.

The *closest point* technique was preferred by the vast majority of users. It received the best rankings in all categories. In terms of intuitiveness it was tied with the *freehand* technique. Users reported that the visible drawing point that the technique put onto the virtual object served as a useful reference point. They also stated that using the *closest point* technique "felt like drawing". Users liked that the *closest point* mode allowed them to draw accurately, even if the pen tip has some distance to the target object. It was also remarked that the *closest point* technique was more robust to mistakes than the other modes.

The *raycasting* mode received the worst feedback from the users. It ranked lowest in all categories. In terms of perceived precision, it was tied with the *freehand* technique. Users reported that they had to think a lot about where the ray was and how it would intersect with the virtual object. Some suggested showing the ray at all times, so that there is a visible cue whether they are hitting or missing the virtual object, and how to adjust the pen position in order to hit the target object. This was not included in the current implementation, since the other drawing modes did not have additional visualization helps either. Although the tracking of the pen was reported as very stable, users noted that there was a larger amount of jittering when using the *raycasting* technique, compared to the others. This can be attributed to the fact that the drawing node is at the end of a longer lever in the *raycasting* mode compared to the other modes, which causes any trembling in the drawing hand or noise in the tracking to be amplified. On the other hand, some users liked using the *raycasting* technique on the front-facing sides of the cube, since it allowed them to draw "from the wrist" instead of having to move the entire arm.

The *raycasting* mode was disliked by most users.

The *raycasting* mode might benefit from better visualization.

6.2 Quantitative Results

To evaluate the drawing performance, we recorded the pen tip's position. Since the guided drawing modes use a different drawing node that is projected onto the virtual cube's surface, we recorded this node's position as well. We also tracked the virtual study object's position in the case the model had drifted mid-trial. These measures were recorded for each frame in which the user was pushing the drawing button. We used the data of 380 completed trials for the analysis.

From these measures we calculated the drawing precision for each frame. That is, we calculated the Euclidean distance between the drawing node's position and the nearest point on the target line. For this we employed the same algorithm used for the *closest point* drawing mode.

The mean Euclidean distance to the target line was our primary metric.

We then performed an ANOVA test on Aligned Rank Transformed data. For this, we averaged the drawing precision for each side of the task cube, for each trial. The conditions we investigated are shown in Table 6.1, the resulting p-values are shown in Table 6.2.

Variable	Condition
Drawing mode	*closest point* *raycasting* *freehand*
The location of the target line on the cube (*CornerType*)	*middle* *interior* *up* *right* *front*
Size of the target cube	*small*: 8cm *large*: 12cm
The side of the cube the user is drawing on	*front* *back* *right* *left* *top* *bottom*

Table 6.1: Variables and conditions for the statistical analysis.

Considering the overall drawing precision (see Figure 6.2), it appears that the guided drawing modes *closest point* and *raycasting* resulted in more precise results than the freehand mode. However, since there appear to be many significant interaction effects between the different conditions, we examined the results more closely.

The guided drawing modes perform best when not drawing around corners.

Examining the conditions "DrawingMode x CornerType" (see Figure 6.3), we can see that both *closest point* and *raycasting* performed noticeably better in the conditions "up", "right" and "front". This is likely due to the fact that the target line for those trials is located along the edges of one side of the cube. Participants could solve these trials without having to re-orient themselves around the different corners,

No.	Conditions	p-value
1	Size	0.01876110
2	Side	4.9692e-11
3	CornerType	$< 2.22e-16$
4	DrawingMode	$< 2.22e-16$
5	Size x Side	0.68346223
6	Size x CornerType	3.8797e-05
7	Side x CornerType	1.3038e-06
8	Size x DrawingMode	0.70108630
9	Side x DrawingMode	6.9500e-09
10	CornerType x DrawingMode	$< 2.22e-16$
11	Size x Side x CornerType	0.00012694
12	Size x Side x DrawingMode	0.41000384
13	Size x CornerType x DrawingMode	0.33230841
14	Side x CornerType x DrawingMode	3.1212e-10
15	Size x Side x CornerType x DrawingMode	0.00511080

Table 6.2: Results of the ANOVA test.

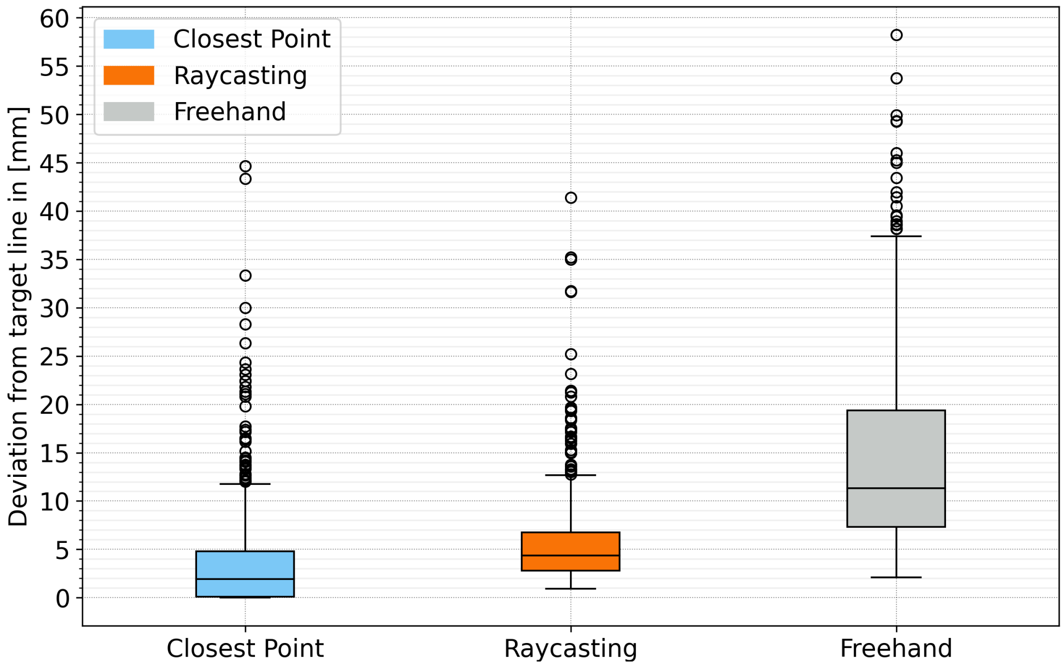

Figure 6.2: Overall drawing precision for each drawing mode. Each data point is the mean 3D deviation from the target line of one side of one trial.

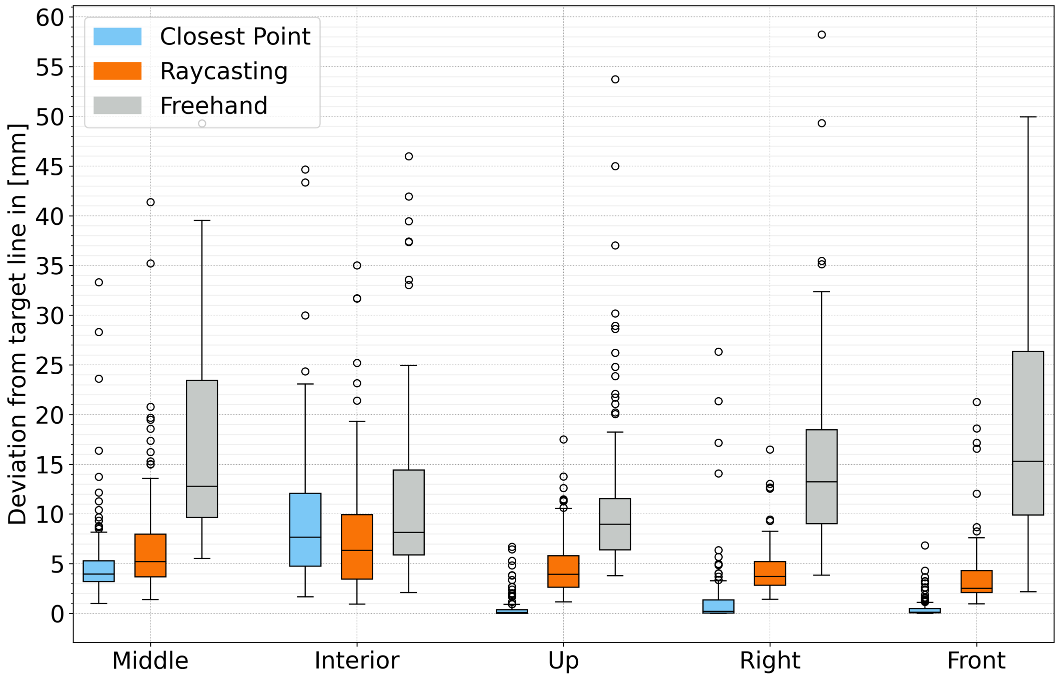

Figure 6.3: Drawing precision for the conditions "DrawingMode x CornerType".

and without having to draw on the back-facing side of the cube, as they had to do in under the conditions "middle" and "interior". Especially the *closest point* mode performed almost perfectly in these conditions, since it is remarkably easy to trace exterior edges in this mode.

The *closest point* mode performs worse on concave geometry.

Also noteworthy is the performance of the *closest point* mode in the "interior" condition. This performed much worse than the other conditions because on interior corners there are often two different valid surface positions between which the *closest point* guidance switches frequently. A possible solution to this is introducing a threshold amount, under which the drawing node sticks to one object, even if there is another valid object nearby.

The freehand mode shows a relatively uniform performance throughout most conditions. The precision seems to improve under the "interior" condition. This may be due to the fact that the cube in this condition is separated by a plane, which can serve as a depth cue that is missing in the other trial conditions.

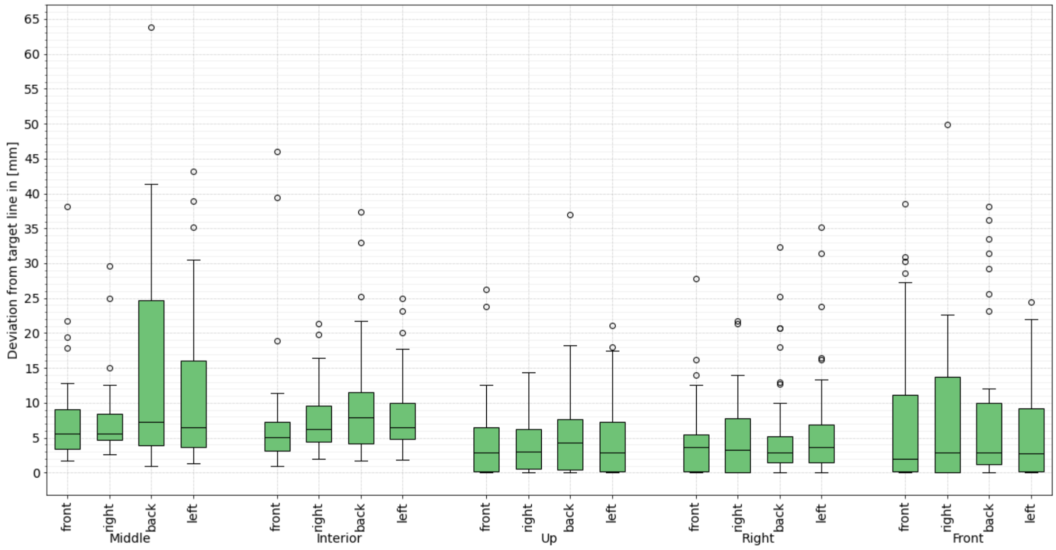

Figure 6.4: Drawing precision for the conditions ”Side x CornerType”.

Examining the conditions ”Side x CornerType” shows that drawing precision on the back-facing side of the ”middle” and ”interior” type cubes tends to be worse for a significant portion of users. This can be explained by the fact that the users have to draw on a side of the cube where they have fewer depth cues than on the front side. Although the cube is semi-translucent, the change in color on the virtual pen tip was used by the participants to check whether the pen tip was in front of the cube, within the cube or right on the edge. This was not possible while drawing on the back-facing side of the cube. Notably, some users leaned right, to draw on the right side of the cube, but did not lean left, when drawing on the left side, possibly because they are right-handed. In the ”middle” condition, this caused them to draw without sufficient depth cues, while working on the left side of the cube. This effect is not observed in the ”interior” condition. This may be because the plane separating the cube along the middle served as an additional depth cue.

While drawing on the back-facing side of the cube, a lack of depth cues becomes an issue.

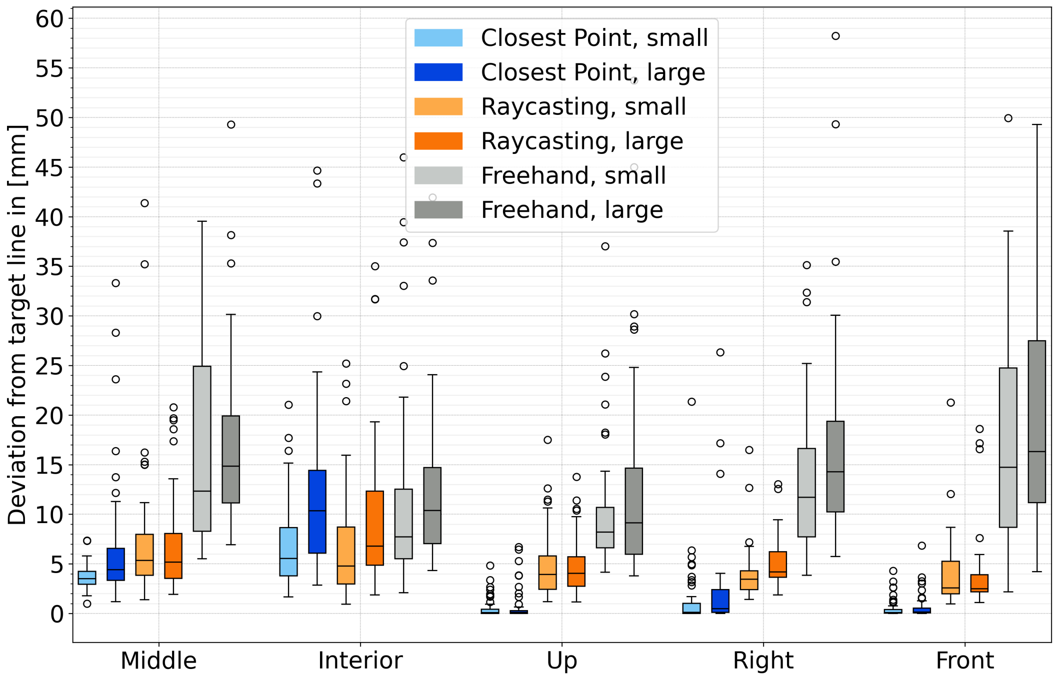

Figure 6.5: Drawing precision for the condition "Size x CornerType".

The impact of the object *size* does not seem to be strong (see Figure 6.5). The only noteworthy difference seems to be that the *closest point* drawing mode performs worse on the *large interior* condition compared to the *small interior* condition. Also, the *freehand* mode seems to perform slightly better in the *small* condition than in the *large* condition.

Chapter 7

Summary and Future Work

7.1 Summary and Contributions

We implemented and evaluated two guided drawing modes.

First, we gave an overview of different studies and projects relating to sketch-based modeling. Afterwards we designed and implemented two techniques for guided 3D drawing in Augmented Reality using the ARPen. The two drawing modes are *closest point*, where the drawing node of the pen is moved to the nearest surface location on a nearby virtual object, and *raycasting*, where a ray is projected from the pen tip and the drawing node is placed at the first intersection of that ray with a virtual object.

To evaluate the drawing techniques we conducted a user study during which participants used the two guided drawing modes and the default freehand drawing mode to trace lines on virtual cubes. Participants preferred the *closest point* mode strongly over both the *raycasting* and the freehand drawing modes. The drawing precision of the two guided drawing modes was better than that of the freehand mode.

Both guided drawing modes improve the drawing precision.

Overall, both guided modes performed better than the freehand mode across most conditions. While the *closest point* drawing mode performs better on flat or convex geometry in terms of precision, it has issues when drawing on concave geometry, since there are multiple possible nearest points under those conditions and the guidance jumps between those. In contrast, the *raycasting* mode performs well on flat or concave geometry, but there is a significant amount of tremor in the drawing, especially when the pen is held at a large distance from the object surface or when drawing at a sharp angle. Thus users should have the option to switch between drawing modes to select the technique that best suits the situation at hand. Most participants of the user study gave overall positive feedback for the *closest point* mode and more critical feedback towards the *raycasting* mode, with the freehand mode ranking somewhere in the middle.

Most participants of the study preferred the *closest point* mode and disliked the *raycasting* mode.

7.2 Future Work

The *raycasting* mode might be improved with better visualizations.

While the results of our user study show *closest point* mode to be favored by users, and *raycasting* to be disliked, implementing more visualization for the *raycasting* mode — particularly displaying the ray — may lead to more favorable results for the *raycasting* mode. Moreover the study tasks may have favored the *closest point* technique over *raycasting*. Thus, in real use cases the conditions where *raycasting* performs equally well or better than *closest point* may be more prevalent than in our study. Similarly the *closest point* mode may benefit from better visualization of the pen tip location.

The "raycasting from the camera" concept may be useful for pointer interactions.

A drawing technique that we considered but did not implement was a "raycasting from the camera" mode, as described in Section 4.4. The reasoning against this drawing technique was that it does not allow the user to draw on back-facing geometry, and that, in a sense reduces the 3D interaction of freehand drawing to a 2D pointer interaction on the screen. In combination with now implemented rotation techniques for the ARPen, this drawing technique may be a viable alternative to the two modes evaluated in this thesis.

While advanced modeling techniques for the ARPen have been implemented ([Benscheid, 2019]), the ARPen may lend itself well to the sketch-based modeling techniques of ”Teddy” ([Igarashi et al., 2006]) and ”FiberMesh” ([Nealen et al., 2007]). Since the techniques presented in this thesis improve drawing precision compared to freehand 3D drawing, implementing modeling techniques that rely on drawing on surfaces, such as Teddy or FiberMesh would make sense.

Sketch-based modeling techniques such as in ”Teddy” and ”FiberMesh” might work well with the guided drawing modes.

Appendix A

Questionnaire

In this Appendix we show the questionnaire given to the participants of our user study. The questionnaire was sent to the users as a Google Form document.

User Study Questionnaire

Questionnaire about drawing guides for the ARPen

* Required

1. What is your age?

2. What gender do you identify as?

   Mark only one oval.

   - Female
   - Male
   - Other
   - Prefer not to say

3. How much experience do you have with Augmented Reality (AR)?

   Mark only one oval.

	1	2	3	4	5	
I have never used it						I use it regularly

4. How much experience do you have with drawing, sketching or painting?

   Mark only one oval.

1	2	3	4	5

5. How much experience do you have with digital 3D modeling?

Mark only one oval.

1	2	3	4	5
⬭	⬭	⬭	⬭	⬭

6. How much experience do you have with 3D printing and personal fabrication?

Mark only one oval.

1	2	3	4	5
⬭	⬭	⬭	⬭	⬭

7. How much do you agree with the following statements?

Mark only one oval per row.

	Strong disagree	Disagree	Neutral	Agree	Strong agree
I liked using the ARPen	⬭	⬭	⬭	⬭	⬭
I found the ARPen intuitive to use	⬭	⬭	⬭	⬭	⬭

8. How would you rank the three drawing modes used in this study overall? *

Mark only one oval per row.

	Worst	Middle	Best
Closest point drawing	⬭	⬭	⬭
Raycast drawing	⬭	⬭	⬭
Freehand drawing	⬭	⬭	⬭

9. Which drawing mode did you find most intuitive? *

Mark only one oval per row.

	Least intuitive	Middle	Most intuitive
Closest point drawing	⬭	⬭	⬭
Raycast drawing	⬭	⬭	⬭
Freehand drawing	⬭	⬭	⬭

10. Which drawing mode did you find most precise? *

Mark only one oval per row.

	Least precise	Middle	Most precise
Closest point drawing	⬭	⬭	⬭
Raycast drawing	⬭	⬭	⬭
Freehand drawing	⬭	⬭	⬭

11. Which drawing mode did you find most physically comfortable to use? *

Mark only one oval per row.

	Least comfortable	Middle	Most comfortable
Closest point drawing	⬭	⬭	⬭
Raycast drawing	⬭	⬭	⬭
Freehand drawing	⬭	⬭	⬭

12. What did you like about the drawing guides?

10. Which drawing mode did you find most precise? *

Mark only one oval per row.

	Least precise	Middle	Most precise
Closest point drawing			
Raycast drawing			
Freehand drawing			

11. Which drawing mode did you find most physically comfortable to use? *

Mark only one oval per row.

	Least comfortable	Middle	Most comfortable
Closest point drawing			
Raycast drawing			
Freehand drawing			

12. What did you like about the drawing guides?

Appendix B

User Study Consent Form

Informed Consent Form

Evaluating Virtual Drawing Guidance for Mobile Augmented Reality

Principal Investigator Martin Huppertz
Media Computing Group
RWTH Aachen University
Email: martin.huppertz@rwth-aachen.de

Purpose of the Study: The goal of this study is to evaluate techniques that guide users to draw in Augmented Reality (AR). Specifically, we investigate mobile, handheld-AR with a smart phone and a bluetooth pen that can be used to draw in mid-air. Participants will be asked to trace virtual lines in AR on objects in 3D space, utilizing the various guidance techniques. The positions of the drawing pen in mid-air and the phone's camera feed will be used in the analysis.

Procedure: Participation in the study will involve two phases. In the first phase you will have the opportunity to familiarize yourself with our AR drawing app. In the second phase, you will be asked to trace highlighted lines on virtual objects, using three different guidance techniques.

After the study, you will be asked to fill out the questionnaire. You will be asked about your experience with AR, drawing, 3D modeling, and similar activities and technologies. You will also be asked to evaluate the guidance techniques used in this study. Overall, the entire study should take about an hour to complete.

Risks/Discomfort: You may become fatigued during the course of your participation in the study. You will be given several opportunities to rest, and additional breaks are also possible. Considering the current outbreak of the coronavirus, we take precautions to design this study as safe as possible for everyone involved. However a remaining risk cannot be excluded. There are no other risks associated with participation in the study. Should completion of either the task or the questionnaire become distressing to you, it will be terminated immediately.

Benefits: The results of this study will be useful for developing more intuitive input modes for mobile AR using a phone and pen.

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

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

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

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

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

If you have any questions regarding this study, please contact Martin Huppertz via email:

Bibliography

Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *CHI*, volume 17, pages 5643–5654, 2017.

Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, page 151–160, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939753. doi: 10.1145/1449715.1449740. URL `https://doi.org/10.1145/1449715.1449740`.

Mayra D Barrera Machuca, Paul Asente, Jingwan Lu, Byungmoon Kim, and Wolfgang Stuerzlinger. Multiplanes: Assisted freehand vr drawing. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3, 2017.

Jan Benscheid. Making sense of lines: Interaction sequences for 3d modeling with mid-air sketches. Master's thesis, RWTH Aachen University, Aachen, October 2019.

Alexandra Bonnici, Alican Akman, Gabriel Calleja, Kenneth P Camilleri, Patrick Fehling, Alfredo Ferreira, Florian Hermuth, Johann Habakuk Israel, Tom Landwehr, Juncheng Liu, et al. Sketch-based interaction and modeling: where do we stand? *AI EDAM*, 33(4):370–388, 2019.

Michael Gervautz and Dieter Schmalstieg. Anywhere interfaces using handheld augmented reality. *Computer*, 45 (7):26–31, 2012.

Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM SIGGRAPH 2006 Courses*, pages 11–es. 2006.

Donna Klamma. Rotating objects: Implementation and evaluation of rotation techniques for the arpen system. Bachelor's thesis, RWTH Aachen University, Aachen, August 2019.

Ernst Kruijff, J Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12. IEEE, 2010.

Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007 papers*, pages 41–es. 2007.

René Schäfer. Redesigning arpen: Evaluating different marker placements for mid-air pen interaction. Master's thesis, RWTH Aachen University, Aachen, June 2020.

Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. Physical guides: An analysis of 3d sketching performance on physical objects in augmented reality. In *Proceedings of the 6th Symposium on Spatial User Interaction*, SUI '18, New York, NY, USA, October 2018. ACM. ISBN 978-1-4503-5708-1. doi: 10.1145/3267782.3267788. URL `https://doi.org/10.1145/3267782.3267788`.

Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. Evaluating menu techniques for handheld ar with a smartphone mid-air pen. In *To appear: 22nd International Conference on Human-Computer Interaction with*

Mobile Devices and Services, MobileHCI '20, New York, NY, USA, October 2020a. ACM. ISBN 978-1-4503-7516-0/20/10. doi: 10.1145/3290605.3300849. URL `https://doi.org/10.1145/3379503.3403548`.

Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. Heatmaps, shadows, bubbles, rays: Comparing mid-air pen position visualizations in handheld ar. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 719:1–719:11, New York, NY, USA, April 2020b. ACM. ISBN 978-1-4503-6708-0/20/04. doi: 10.1145/3313831.3376848. URL `https://doi.org/10.1145/3313831.3376848`.

Felix Wehnert. Pen-based drawing in augmented reality on mobile phones. Bachelor's thesis, RWTH Aachen University, Aachen, April 2018.

Min Xin, Ehud Sharlin, and Mario Costa Sousa. Napkin sketch: handheld mixed reality 3d sketching. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 223–226, 2008.

Typeset August 25, 2020