# A Two-Dimensional Input Modality for Extrusion of Custom 2D Shapes with AR-Based Stereoscopic Visualization

Lisa M. Lichtenberger

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2019

# Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 24, 2019

Lisa M. Lichtenberger

# Contents

# Preface

This master thesis was developed in the course of the master's program *Interactive Media* at the University of Applied Sciences Upper Austria and covers topics in which my interest has developed and grown throughout my studies.

To put it in a nutshell, I would like to say that although the last five years have had their ups and downs, I have always been very proud to be able to study at the University of Applied Sciences. I would also like to thank a group of people who supported and accompanied me during my studies and without whom I could not have achieved my goal.

My supervisor for this thesis, FH-Prof. Dr. Christoph Anthes MSc, supported me a lot in finding and specifying the topic for my master thesis. I also want to thank him for the constant support during the development process in form of numerous meetings, helpful tips and ideas, constructive feedback, and also the kind working atmosphere.

Furthermore, I want to thank my parents, who always stood by me and strengthened my back in every situation, especially my mother Erna Lichtenberger and my twin sister Theresa S. Lichtenberger.

What I am also very grateful for are my fellow students (no matter which study program) which have become very good friends for me and in particular, my flatmates with whom I enjoyed nearly four years of living, celebrating and studying in Hagenberg.

# Abstract

Traditional paper sketching is fast and intuitive. Finding a way to sketch 3D objects as natural as with paper sketching is indeed an ambitious concern and using AR technologies may be a way to enhance sketching objects in 3D space. The research in this field is pervasive, and there are research groups that already investigated the effect on using AR and stereoscopic visualization in combination with object creation and manipulation and the resulting advantages and disadvantages of it.

The focus of the thesis is on object creation and manipulation in conjunction with AR technologies. In more detail, this means that users are creating objects by drawing custom 2D shapes, extruding them along a custom extrusion path, manipulate the created objects with basic object transformations (translation, rotation, and scaling), and use a planar as well as a stereoscopic visualization for viewing the created objects.

The main objective was to find out if a stereoscopic visualization is supporting the user in creating objects and also to find advantages and disadvantages when working with stereoscopic visualization.

Therefore, a prototype has been developed which provides a user interface for sketching objects with extrusion, manipulating objects and also offers a planar and stereoscopic visualization of the generated extrusion objects. A methodical approach has been organized and realized step by step, which included conceptual design, technical implementation, and the evaluation, which included a quantitative as well as a qualitative evaluation.

# Kurzfassung

Eine herkömmliche Art und Weise, um schnell und intuitiv eine Idee einzufangen und anderen Menschen zu präsentieren, ist, sie mit einem Stift auf einem Blatt Papier zu skizzieren. Um jedoch dreidimensionale Objekte zu skizzieren, bedarf es etwas an Erfahrung, Können und einem räumlichen Vorstellungsvermögen, welches womöglich durch die Verwendung von Augmented Reality Technologien erleichtert oder sogar verbessert werden könnte. Die Forschungsarbeit auf diesem Gebiet ist mittlerweile sehr umfangreich und Forschungsgruppen haben bereits im Zuge von einigen Projekten die Auswirkung von Augmented Reality Technologien und stereoskopischer Visualisierung im Zusammenhang mit Objekt Generierung und Manipulation untersucht.

Der Fokus dieser Arbeit liegt auf der Erstellung eines Prototyps mit welchem Benutzer in Verbindung mit Augmented Reality Technologien Objekte sowohl erzeugen als auch manipulieren können. Genauer genommen erzeugt der Benutzer Objekte durch das Extrudieren von zuvor gezeichneten, zweidimensionalen Formen. Der Benutzer bestimmt somit sowohl den Grundriss des Objektes als auch den Extrusionspfad. Nach der Erzeugung des Objektes kann der Benutzer das Objekt mit etablierten Objekttransformationen (Verschiebung, Rotation und Skalierung) manipulieren und das Objekt sowohl als planare Darstellung auf einer tablet-basierten Oberfläche als auch als stereoskopische Darstellung mit Hilfe einer Mixed-Reality-Brille betrachten.

Das Hauptziel der Arbeit war es, herauszufinden, ob eine stereoskopische Visualisierung dem Benutzer während der Objekterzeugung und Manipulation behilflich ist und welche Vor- und Nachteile aus einer stereoskopischen Visualisierung resultieren.

Um dieses Ziel zu erreichen, wurde ein Prototyp entwickelt, welcher eine Benutzeroberfläche zur Verfügung stellt, mit dessen Hilfe ein Benutzer Objekte durch Extrusion erstellen, diese Manipulieren und sie mittels planarer und stereoskopischer Visualisierung betrachten kann. Hierfür wurde eine methodische Vorgehensweise gewählt, welche aus dem konzeptuellen Design der Benutzeroberfläche und den Benutzereinteraktionen, der technischen Implementierung und einer anschließenden Auswertung mit Hilfe einer Benutzerstudie bestand und Schritt für Schritt ausgeführt wurde.

# Chapter 1

# Introduction

A common way to present a concrete idea to someone else is to sketch the idea with a pen on paper. Paper sketching is a fast and intuitive option for illustrating ideas, capturing the personal vision, and also to make the own vision clear to other persons. In order to sketch three-dimensional objects, however, it takes some experience, skill, and spatial imagination, which could be facilitated or even improved through the use of *Augmented Reality* technologies. *Augmented Reality* has gained much attention over the last decades, and the possibilities offered by its technologies are continually being researched, explored, and further developed. Of course, the stereoscopic representation of objects can also serve as an additional aid during the creation process of three-dimensional objects, scenes, and sketches.

In general terms, a multitude of input modalities, such as with mouse, keyboard, pen, hand gestures or touch input, have been researched in the field of *Human-Computer Interaction*, and there is also a variety of displays which have been developed in the field of *Augmented Reality*, ranging from simple desktop applications to hand-held displays and head-mounted displays, to display virtual data. It is therefore essential to find the best possible combination of input modality and content display in order to achieve a positive effect during the development process of three-dimensional objects.

## 1.1 Motivation

The focus of the thesis is on the extrusion of custom 2D shapes and the stereoscopic AR-based presentation of the extruded 3D objects. Creating 3D objects out of custom 2D shapes is not a trivial task, and early creation of three-dimensional geometries with 2D interfaces requires experience and knowledge. To preserve the simplicity of traditional paper sketching, a tablet-based user interface with which the user can interact and create 3D objects by using a pen would be desirable. Regarding the thesis, 3D objects are created with extrusion and perceived either on the planar and tablet-based surface or as stereoscopic visualizations. Based on *Augmented Reality* (AR), the stereoscopic visualizations are displayed directly in the field of view of the user with a head-mounted display (HMD) the user is wearing during interacting with the interface.

## 1.2 Research Question

The user should be enabled to interact with the 2D interface, and stereoscopic augmentation should be used to visualize the corresponding three-dimensional object which the user created with extrusion. Therefore, the focus of this master thesis will be on the following research question.

> How does AR-based stereoscopic visualization support the user in creating extrusion objects, and what are the advantages and disadvantages of it?

## 1.3 Goals and Methodical Approach

In order to investigate the introduced research question, the following methodological approach was chosen, which consists of three steps: Concept, Implementation, and Evaluation.

### 1.3.1 Concept

The main goal of concept development is to create sketches for the user interface and user interaction of the prototype, which are then implemented.

### 1.3.2 Implementation

The role of the thesis project is not to implement a modeling tool, but to focus on extrusion, amongst the set of possible modeling operations.

The project consists of two applications: one for the tablet-based surface and one for the see-through head-mounted display. Those two applications are communicating with each other over a network connection.

The user draws a custom 2D shape on a planar and tablet-based surface and extrudes this shape by drawing another custom line or path in 2D. Accordingly, the 2D shape will extrude into 3D space along the user-defined path. The extruded 3D objects can be displayed either on the planar surface or as an AR-based stereoscopic visualization directly on the 2D surface. The stereoscopic visualization is placed respectively to a planar image marker with which the position of the tablet-based surface can be recognized and tracked. Additionally, the user can apply fundamental transformations such as translation, rotation, and scaling and can select and delete objects.

### 1.3.3 Evaluation

With the help of a user study, the resulting prototype will be evaluated in terms of advantages and disadvantages when using AR for stereoscopic visualization.

On the one hand, registration of AR-based stereoscopic visualization directly attached to and and placed on the tablet and on the other hand, a conventional planar 3D presentation on the tablet will be compared in a given usage scenario. The user will be given a prepared task, for instance, to sketch a predefined object. The user has to rebuild the object in different scenarios: with AR-based stereoscopic visualization on the tablet-based surface and without stereoscopic but with planar visualization on the tablet-based surface. A quantitative and qualitative evaluation will accompany the user study.

### Quantitative Evaluation

For quantitative evaluation, time and precision will be determined to investigate how much time it takes for the participants to finish a task, the error rate (How often is the undo, redo and delete option executed?) and to rate the precision of the result object the participant has created.

### Qualitative Evaluation

For qualitative evaluation, the perceived workload for participants will be rated with the *NASA Task Load Index* (NASA-TLX), which is a widely used assessment tool. NASA-TLX includes aspects such as mental and physical demands, effort and frustration and asks participants to rate and weight these aspects.

Chapter 2

# Fundamentals and Related Work

This chapter presents publications and related projects which form an important part of the research field of this thesis.

## 2.1 Terminology

In the following section, the term *Augmented Reality* is shortly introduced since it is part of the fundamentals for the topic of this master thesis.

### 2.1.1 Augmented Reality

The term *Augmented Reality* (AR) describes a domain that focuses on embedding virtual objects into the real world. Subsequently, AR enables the user to see the real world and virtual objects coexisting in the same space and is, therefore, part of the *virtuality continuum* described by Paul Milgram [21]. The concept of a virtuality continuum includes pure real and virtual environments as two opposite poles and combines different mixtures of reality and virtuality (*Mixed Reality*) which are arranged along the continuum between these two poles. According to Azuma [2], systems can be considered as AR systems if they feature characteristics such as the combination of real and virtual objects, interaction in real-time and registration in 3D. Those are the essential components of AR.

## 2.2 Core Aspects of the Thesis

The focus of this master thesis is on extrusion of custom 2D shapes and the stereoscopic AR-based presentation of the resulting extruded 3D objects. Therefore, the main aspects of this subject area are researched in the following section.

### 2.2.1 Drawing Custom 2D Shapes

ILoveSketch [3] is a 3D curve sketching system that focuses on natural pen-and-paper-like interaction and is therefore closely related to the current topic since they focus on sketching on a 2D surface using a pen. Features such as automatic view rotation, 2D to 3D navigation, and an intuitive gesture vocabulary have been introduced, which are

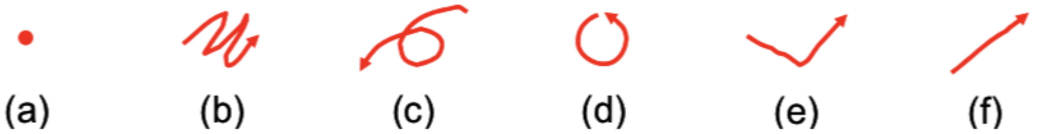


**Figure 2.1:** Gesture set of ILoveSketch which defines drawing patterns for the user using a pen [3].

shown in figure 2.1. The so-called gesture set [3] of ILoveSketch includes some pre-defined gestures which are very similar to classic paper sketching mechanisms. The gesture set consists of patterns that are drawn with the pen by the user.

In terms of pre-defined patterns, a new design for a sketch-based modeling interface by Islam Gharib [13] also contributed patterns for 2D input drawings which are mapped to 3D objects, and for instance, a drawn circle is mapped to a sphere.

In general, some of the requirements and design goals for ILoveSketch are amongst others minimal interruption to sketching by GUI and gestures, a minimal set of intuitive gestures, immediate access to 2D and 3D navigation and dynamic information display to assist 2D and 3D sketching [3].

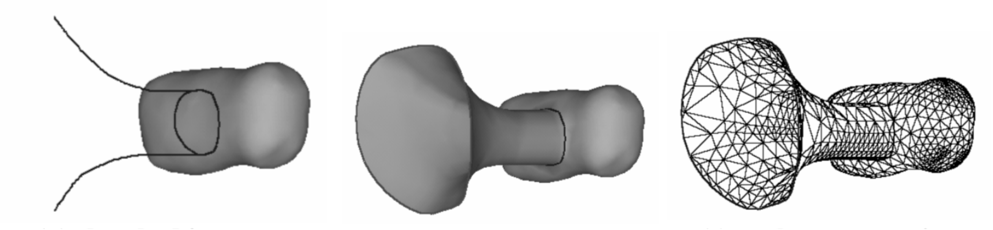
Zabramski et al. [31] also strengthen the assumption of pen interaction being the most accurate one besides touch and mouse input. Therefore, pen interaction is a very appropriate input method for the use case of this thesis because users have to draw lines, and using a pen is also the most intuitive way for drawing.

### 2.2.2 Extrusion

In terms of computer graphics, extrusion refers to generating 3D objects from 2D forms and shapes by essentially stretching out a line of the 2D form along a new third axis. For example, a quadratic form defined within the xy-plane can be extruded and therefore be turned into a cube by using extrusion along the z-axis.

Regarding the project related to this master thesis, to create a 3D object, the user, first of all, has to draw strokes by using a pen. Then, those strokes are interpreted as a 2D shape, and the user can create a 3D object by extruding the previously drawn 2D shape into 3D space which has also been investigated by Eggli et al. [10]. The system, described by Eggli et al. [10], firstly interprets pen strokes as geometric profiles. Afterwards, the system allows extrusion of 2D profiles (previously drawn pen strokes) by sweeping the profile along a straight line or a sketched curve. Pen strokes can also be drawn on existing objects to quickly add new geometry to existing objects, similar to [30] and [16].

SKETCH [32] focuses on combining pen and paper sketching with features of a CAD system to close the gap between hand sketching and full-grown modeling systems by only sticking to a gesture-based interface. SKETCH also features freehand drawing of closed curves and extrusion paths and additionally offers the possibility to attach both endpoints of an extrusion path to two objects so that the extrusion adapts accordingly if one of the objects is moved or transformed.



(a) (b) (c)

**Figure 2.2:** Sketched input (a), Extrusion result (b), Generated Mesh (c). Source: [30]

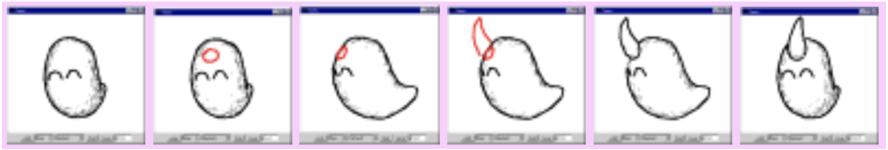


**Figure 2.3:** Extruding a drawn shape by rotating view and drawing extrusion silhouette [16].

A sketch-based mesh extrusion has also been researched and investigated by Wang and Yuen [30]. As presented in figure 2.2, their approach enables users to sketch closed as well as open shapes onto other surfaces or on boundaries of given object meshes. After sketching the shape, the user can extrude them by adding extruding strokes. The mesh extrusion algorithm of Wang and Yuen [30] is divided into four steps which are illustrated in figure 2.2. The first image in figure 2.2 presents the first and second step, which is about the sketched input. The user creates the base curve at first (which is a circle in the image) and then draws so-called extruding strokes afterwards which are, in this case, funnel-shaped. The third step is sweeping the base curve to generate the resulting extrusion object, which is shown in the second image in figure 2.2. The fourth step is to merge the resulting extruded object with the adjacent curves of the original surface or mesh. This approach is also very closely related to the extrusion method of the Teddy [16] system at which point Teddy is limited to extrusion of only closed shapes on meshes. As shown in figure 2.3, the extrusion workflow involves drawing a closed shape onto an object, rotating the view and drawing the extruding stroke or rather the silhouette of the resulting extrusion object. The red line indicates the extrusion mode. According to Igarashi et al. [16], this extrusion method is a sweep operation that generates a 3D object by moving the closed 2D shape along the extrusion lines (silhouette) in which the extrusion direction is always perpendicular to the surface.

SESAME [24, 25] also supports extrusion of 2D shapes along a straight path which is basically along the surface normal. SESAME will be introduced in more detail in section 2.3.1.

ConMan [14] supports a visual programming language with which users can define a program by using two dimensional notations instead of one dimensional instances such as streams of characters. As shown in figure 2.4, users can create and modify components which can be connected with each other and users can change these connections with

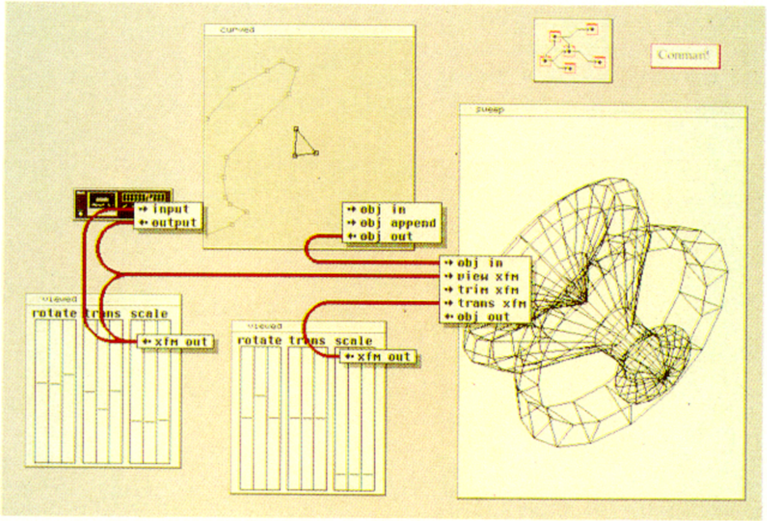


**Figure 2.4:** Example image of user interface of the ConMan system [14].

the connection manager (ConMan). Hence, ConMan has a structured format where separate components describe how a shape is going to be transformed. For example, the curved component (which contains the 2D shape) in figure 2.4 is connected with a sweep component which is again connected with two viewed components that define the transformation that is used to create the swept or rather extruded object. For example, an extruded object can be generated by rotating the 2D shape around the x or y-axis or by translating it in the z-direction.

A different approach to constructing 3D objects out of 2D shapes has been investigated by Shum et al. [27] and Zi-Gang et al. [11]. This approach is nested in the field of reverse engineering and reconstructs solids from orthographic views using incremental extrusion. As shown in figure 2.5, six orthographic views are extracted from a 3D object with camera images which are interpreted as 2D line drawings. Those views need to be opaque, which means that the 2D drawing only contains visible edges of the opaque object. The next step is to group all six views into three pairs and continue with incremental extrusion which is illustrated in the second image of figure 2.5. Each pair consists of two views: the so-called generatrix view and the directrix view. The generatrix view is the one which will be extruded along its adjacent directrix view.

As a result, a primitive solid is created, and this process will be repeated with the other two pairs as well. Finally, all solids, which have been created incrementally by extrusion, are merged into one resulting extrusion object which is shown in figure 2.6. According to Zi-Gang et al. [11], constructive solid geometry (CSG) based approaches can be used to generate the resulting solids with Boolean operations which are applied on the primitive solids.

### 2.2.3 Stereoscopic 3D Visualization

The planar visualization of 3D objects is a popular way to present three-dimensional content to the user since it is known, for example, from CAD systems, video games, or

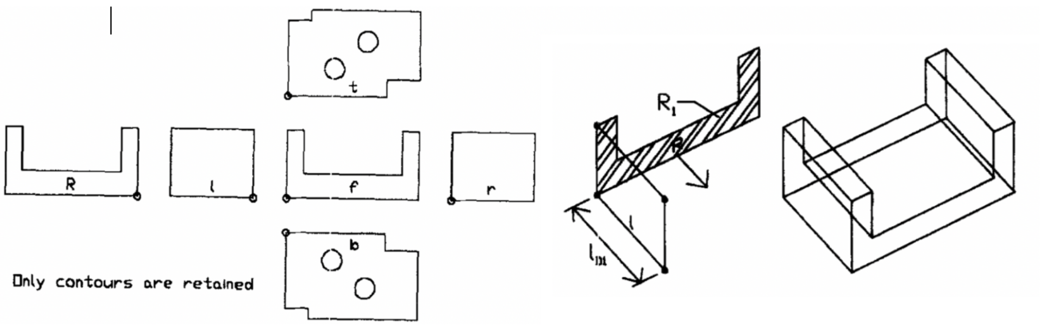


**Figure 2.5:** 3D solid reconstruction with the use of six orthographic views of an object [27].

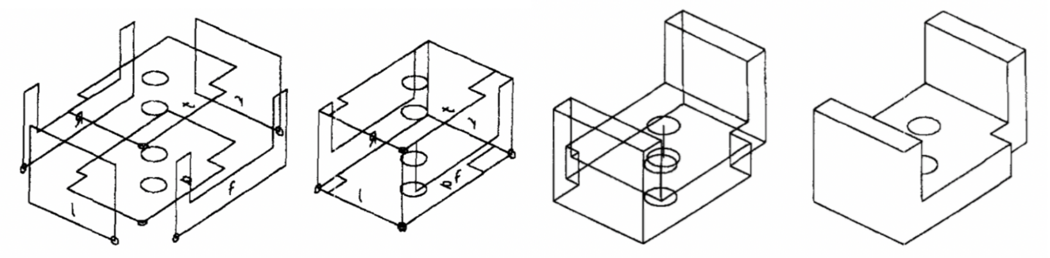
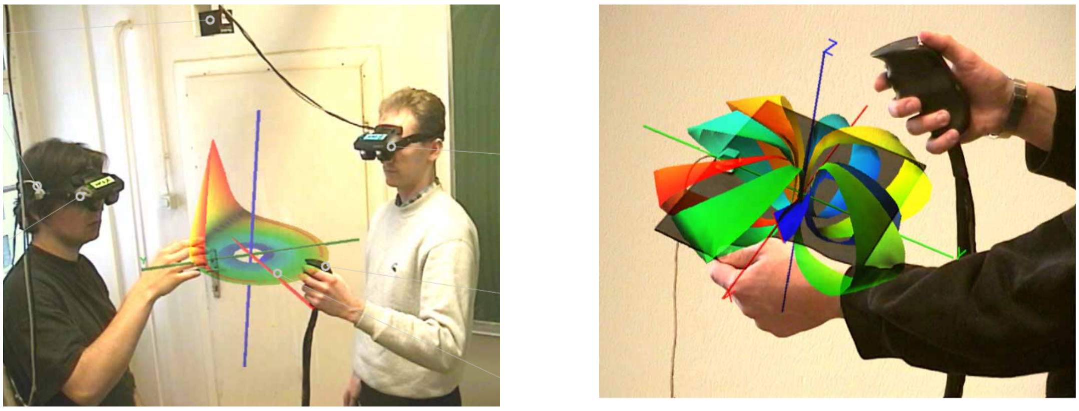


**Figure 2.6:** Extrusion solids are merged to a final solution object [27].

television. According to Bruder et al. [7], a monoscopic (both eyes perceive the same image) presentation of 3D content has a negative impact on the spatial perception of 3D scenes as well as on the performance when completing tasks in 3D space due to the inferior perception. Another option for presenting spatial content to the user is with stereoscopy.

Stereoscopic displays, as stated by Valkov et al. [29], present a stereoscopic visualization of a 3D scene by showing two slightly different images of the 3D scene separately to the left and right eye of the user. The human brain is capable of merging these two images to produce an illusion of depth when viewing an image [19]. Therefore, stereoscopy is a way to enhance spatial understanding of 3D scenes, as stated by McIntire et al. [20], because users showed a clear benefit when it comes to understanding complex 3D scenes and their readability and also when manipulating 3D objects while familiar depth cues [4], for example, shadows, occlusion, and texture, are also applied. Bruder et al. [7] mentioned that there are also drawbacks when it comes to stereoscopic visualization. The stereoscopic visualization may appear detached from the screen, which could be an undesired aspect and that the stereoscopic perception of the object may be impaired depending on the focus of the viewer, for example, blur and having double vision when focusing on the 2D user interface which is behind the stereoscopic visualization [7].

Nevertheless, stereoscopy has become very popular in various application areas, for instance in surgery, scientific visualization and the entertainment sector including the



(a) (b)

**Figure 2.7:** Hardware setup (a) and scientific stereoscopic visualization presented on the Personal Interaction Panel (PIP) (b). Source [12]

3D cinema and television [4]. In a survey of AR [5], two approaches have been compared to combine the real environment with virtual objects which are on the one hand optical and on the other hand video see-through head-mounted displays (HMD). Both have trade-offs regarding resolution, simplicity, registration, temporal delay, depth occlusion, and field of view. A significant advantage of an optical HMD over a video-based optical HMD is the resolution because the user is viewing the real world with his own eyes which can hardly be outplayed by technology. Furthermore, the simplicity of an optical see-through HMD as another benefit since it is more basic and cheaper to achieve than video blending. Problems with the see-through HMD occur especially regarding registration of virtual objects in the real world, temporal delay compared to a video-based HMD and depth occlusion [5].

Studierstube [12] is an example of using multiple optical see-through HMD for a collaborative AR system setup. Their approach is to let multiple users study 3D scientific visualizations simultaneously and in a collaborative way which means that they also can work together on visualizations at the same time.

The figure 2.7 shows the hardware setup of Studierstube. The setup includes an HMD for augmentation, a 3D mouse as the input device for each user in the system as well as magnetic trackers.

Users can also interact with the virtual objects or the stereoscopic visualization itself by using a panel and a pen which allows interaction in 2D as well as 3D and which is tracked by the HMD [1]. The panel is pictured in figure 2.7 and Studierstube used AR to build a 3D user interface that is presented to the user on this panel. The hand-held panel is also called *Personal Interaction Panel* (PIP) which is similar to Construct3D [18].

As described by Gervautz et al. [28], there are different forms of the PIP which range from an intelligent and pressure-sensitive display and a pen to a simple flat panel and a pen where the menu is augmented onto the panel.

Amongst others, Studierstube features customized views which means that a user can choose to see lines added to a visualization whereas another user hides them. Hence,

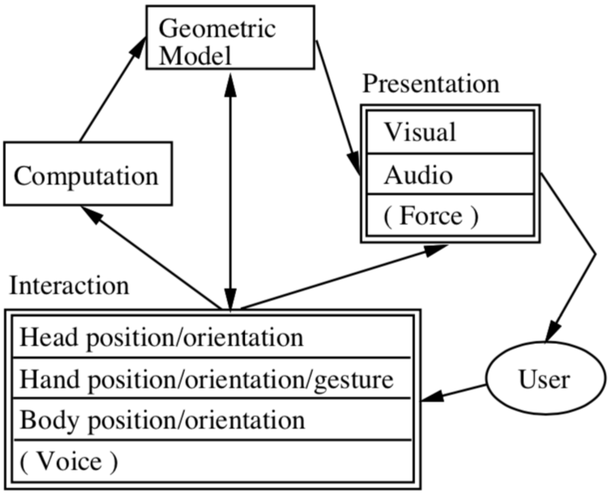

Geometric Model
Presentation
Visual
Audio
( Force )
Computation
Interaction
Head position/orientation
Hand position/orientation/gesture
Body position/orientation
( Voice )
User

**Figure 2.8:** *Decoupled Simulation Model* for structuring Mixed Reality applications [26]

two users can see different versions of the same visualization at the same time which can be very helpful in a collaborative working process.

Regarding collaborative AR and stereoscopic visualization, Construct3D [18] also provides color-coding which marks the contribution of each user participating in the working group with a separate color. Construct3D [18] also implements features such as transparency of stereoscopic 3D visualizations and a layering system to enhance working with augmented 3D geometry and are explained in more detail in section 2.3.2.

Studierstube [12] also uses a slightly simplified version of the so-called *Decoupled Simulation Model* which has been described by Shaw et al. [26] and which is illustrated in figure 2.8.

The *Decoupled Simulation Model* is a well-structured architecture form for applications in the field of Mixed Reality. As shown in figure 2.8, the model consists of four components: the Interaction, the Computation, the Presentation, and the Geometric Model component. The arrows which are drawn between these components represent the data flow within the system.

The Computation component is responsible for non-graphical computations and controlling tasks and therefore, forms the core of the application system. Usually, the Computation component operates independently of other components and continuously outputs the current state of the Geometric Model component.

The Interaction component is listening to any form of input given by the user which can be for example the current head position tracked by an HMD or the current hand position, orientation or gesture similar to the Mockup Builder [17] (see section 2.3.3) system which is using depth cameras and finger tracking to fetch input by hand movement. The voice can also be used as input.

The input data is forwarded to the Computation component, which processes the input data. When the Computation component has finished with processing the new data, it asynchronously updates the Geometric Model component.

Hence, the Geometric Model component maintains a representation of the computed data and gets presented by the Presentation component which displays the current model state and shows it to the user by audible (sound), haptic (vibration) or visual (display) odds. The Geometric Model component can, as well as the Computation component, receive input data directly from the Interaction component because not every user input requires computations.

Concerning the thesis project, this model can also be applied. The user interacts with a surface using a pen. For example, when the user is drawing a line, the input data is processed and forwarded to the Computation component. In the Computation component, amongst others, the mesh is computed, generated, and stored in the Geometric Model component. The current representation of the Geometric Model is transmitted to the HMD, which contains the Presentation component and presents the computed mesh as a stereoscopic visualization to the user. As shown in figure 2.8, various Mixed Reality applications also support direct interaction with the virtual environment or rather the stereoscopic visualization itself, which is not planned for the current prototype of the thesis project.

## 2.3 Overview of Computer-Aided Sketching Tools

Enabling users to interact in a way that feels as natural as possible is one of the oldest problems in the field of computer-aided sketching [25]. Over time, different approaches emerged to improve natural interaction for sketching 3D models. Several sketching tools, sculpting tools, tangible interfaces, and immersive modeling tools exist to simplify sketching in the early development process [23]. In this chapter, several papers will be introduced which focus on aspects that are relevant in the research field of the thesis topic. For the thesis and the related work, three aspects are considered: drawing custom 2D shapes, extrusion, planar, and stereoscopic visualization of 3D objects.

### 2.3.1 SESAME

*SESAME* (Sketch, Extrude, Sculpt, and Manipulate Easily) is a system that supports the creation of general types of shapes by providing 2D drawing capabilities and an extrusion operation. The natural object motion and the directional dragging technique are implemented in this system to support the easy manipulation of a scene.

*SESAME* [24, 25] enables users to draw lines, arcs and also free forms on a planar surface and then let them extrude closed shapes into 3D space. They focus on the presentation of 3D objects on a planar surface. Nevertheless, the presentation of 3D objects on a planar surface is also interesting regarding this thesis since the planar presentation of 3D objects will be compared with a stereoscopic presentation with a see-through HMD.

#### User Interface

The user interface of SESAME, which is shown in figure 2.9, includes a main 3D scene view, which presents the resulting 3D model to the user, and a menu panel which is placed on the right side. In the menu panel, users can select primitive 2D shapes such

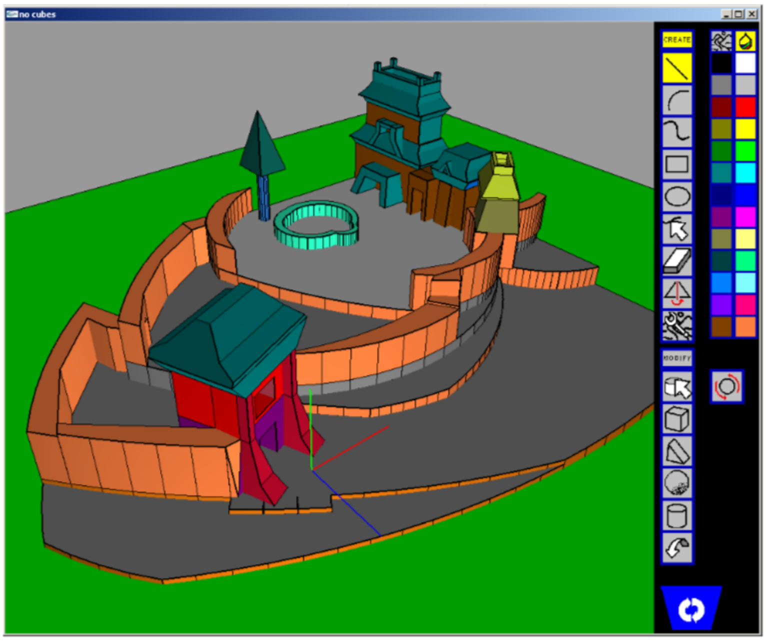


**Figure 2.9:** User interface of SESAME [25].

as lines, arcs, rectangle or free forms as well as 3D shapes such as a cube, cylinder, or sphere. The menu also provides a color and texture palette, an eraser, and an undo button, a recycle bin and a navigation mode switch button. By selecting either a 2D or 3D shape in the menu panel, the user enters the corresponding 2D or 3D mode. The system interface utilizes a 3-button mouse and a few modifier keys to perform all actions such as sketching a contour or moving objects. If the user selects one of the 2D shape tools such as lines, arcs or erasers, then the user enters the 2D mode and manipulates 2D shapes in this mode using the left mouse button and modifier keys. If the user selects one of the 3D object tools, the user enters the 3D mode and manipulates 3D objects (move, add) using the left mouse button and modifier keys again. Additional mouse button functions are independent of the selected mode.

### Extrusion

If the user wants to create a new object, he can either choose a 3D primitive in the toolbar and drag it into the 3D scene view or selects a 2D primitive and draws a 2D shape. If the 2D shape is closed, the system is going to detect it, and it will become possible to extrude the shape into the third dimension. Therefore, the user has to drag the 2D shape from the inside out.

To manipulate the extruded object, the user has two options: add new volume or subtract volume from an existing object by extrusion. As illustrated in figure 2.10, the user draws a 2D shape either freely or on an existing 3D object and extrudes the shape either outwards or inwards the object. The red line in figure 2.10 indicates the path of extrusion. This way of object manipulation offers a flexible way of forming and sculpting an object and is also very intuitive.

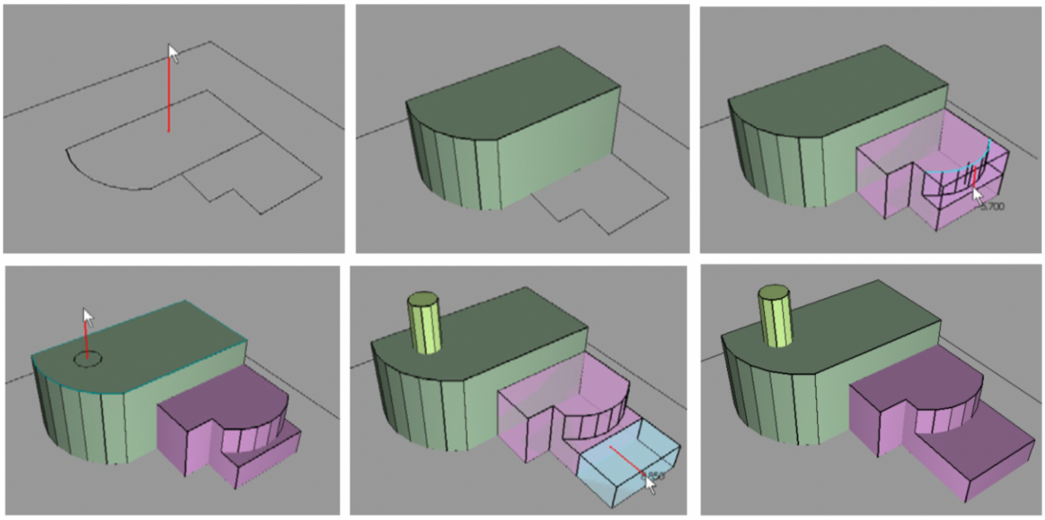


**Figure 2.10:** Extrusion operation in SESAME [25]

### 2.3.2 Construct3D

With *Construct3D*, Kaufmann and Schmalstieg [18] evaluated a collaborative AR tool which has been created to ease geometry education using an immersive setup. By using see-through HMDs for presenting stereoscopic 3D objects and geometries, users or rather the students can see 3D objects from arbitrary perspectives which they previously had to calculate first and than construct them with conventional methods for instance with paper and pen.

The goal of Construct3D [18] was to find out which impact it has to build and create 3D objects directly in 3D space using AR in terms of spatial comprehension of complicated spatial structures. Since Construct3D is meant to be a geometry education software and not a fully grown computer-aided design (CAD) system, they relied on an intuitive interface with a focus on the construction process with simple geometric primitives and operations.

#### User Interface

As shown in figure 2.11, the user interface and accordingly, the menu of Construct3D, is displayed or rather augmented on a so-called Personal Interaction Panel (PIP). The students are interacting with this user interface by using a pen and also are enabled to interact with the stereoscopic visualization directly in 3D space. Construct3D is also very concerned to keep the menu system simple compared to complete CAD systems which typically come with deeply nested menus and a vast number of features. The figure 2.11 also presents the menu which consists of just a few 2D interface components such as buttons, sliders and dials [18]. All features of the system can be accessed easily and fast in contrast to a CAD system.

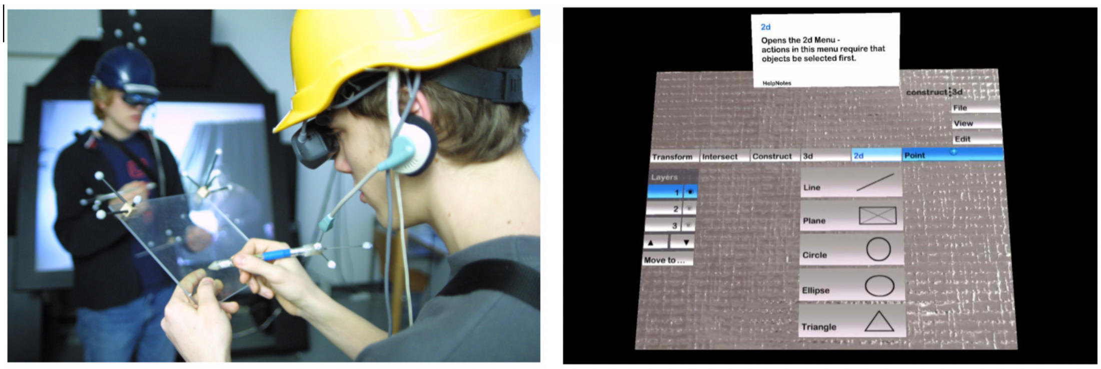


**Figure 2.11:** User interface of Construct3D [18]

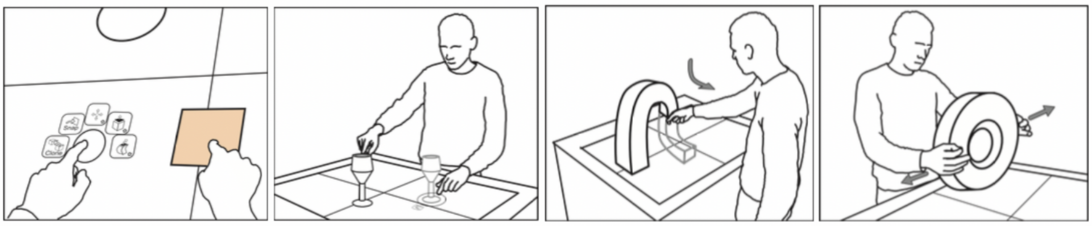


**Figure 2.12:** User Interface and interaction with Mockup Builder: Extrusion along a path and basic object transformations such as object scaling by using both hands [17].

### Visual Design

Construct3D also focuses on improving usability with the help of visual design because the stereoscopic visualization which is displayed by a see-through HMDs cannot be presented with colors that are as bright as on a Desktop or with a closed view HMD. Developers have to face limited contrast when working with HMDs. Furthermore, it is also necessary to deal with limited resolution and viewing angle [18]. Due to this circumstance, it is more difficult for the user to perceive the visualization correctly. Therefore, Construct3D added features such as transparency of objects to enhance the understanding of one or more 3D objects, color coding to tag the contribution of each working partner and a layering system to support semantic structuring.

## 2.3.3 Mockup Builder

Creating and manipulating 3D objects in a continuous interaction space, has been investigated by De Araújo et al., combined with their application Mockup Builder [17]. In this context, continuous interaction space is an environment in which users can seamlessly switch between an interaction on the surface or in the space above the surface. How users need to interact with the system is dependent on the task whether if they create a shape by drawing directly on the surface or extrude the drawn shape along the path, which is defined by hand movement.

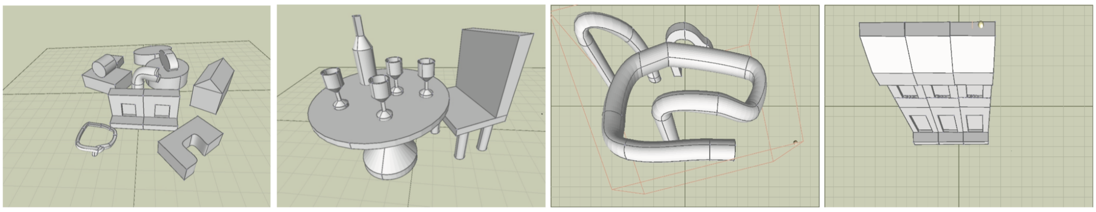


**Figure 2.13:** 3D objects created by using Mockup Builder [17].

### User Interface and Interaction

The user interface and user interaction of Mockup Builder can be considered as semi-immersive since the setup of the application is targeted on providing intuitive user interaction and direct modeling techniques. For the application setup, depth cameras, multi-touch surfaces, and 3D finger tracking devices are used to obtain different data from different sensors which will be combined, analyzed, and interpreted afterwards. Hence, sensor data is chosen accordingly to the current task the user wants to perform to achieve a fluent transition between 2D and 3D interaction. For example, as shown in figure 2.12, different sensor data enables object manipulation by using both hands because the system tracks the hands, and the surface allows multiple inputs simultaneously.

The system allows bimanual interaction and accordingly assigns tasks to the left as well as to the right hand. In figure 2.12, the user interaction with the menu is presented in the first image. The user is selecting a shape with his right hand, and the left hand operates with the menu and selects the appropriate manipulation operation. In this specific case, a multi-touch surface is essential.

In the fourth image in figure 2.12, users can, for example, scale an object by moving their hands apart or towards each other, which is a very natural movement for scaling objects. Subsequently, such a setup establishes advantages that are on the one hand direct user interaction in real-time and on the other hand the close connection between the user and their design ideas because the cognitive overhead is minimized by offering an intuitive workflow.

### Extrusion

Mockup Builder supports drawing of 2D shapes onto a planar surface and extrusion of the resulting shapes into a stereoscopic environment which means that extruded 3D objects are presented in 3D space as stereoscopic visualizations. To realize such a system, Mockup Builder uses depth cameras combined with finger or rather hand tracking. The third image in the figure 2.12 shows how the user can easily extrude a drawn 2D shape into 3D space. The extrusion path is following the movement of the right hand and is defined in this way. In figure 2.13, several 3D objects are presented which have been created with Mockup Builder. Mockup Builder also uses the push and pull manipulation, which can be found in SESAME [24, 25] as well.

| | SESAME | Construct3D | Mockup Builder | Studierstube | Teddy | ConMan | ILoveSketch | SKETCH |
|---|---|---|---|---|---|---|---|---|
| MR continuum | − | AR | AR | AR | − | − | − | − |
| stereo. vis., HMD | − | + | + | + | − | − | − | − |
| collaborative system | − | + | − | + | − | − | − | − |
| input device | mouse | pen | touch, finger tracking | pen, 3D mouse | pen | mouse | pen | three-button mouse |
| PIP | − | + | − | + | − | − | − | − |
| freehand drawing | + | − | + | + | + | + | + | + |
| predefined shapes, patterns, gestures | + | + | + | + | − | − | + | + |
| General extrusion | + | − | + | − | + | + | + | + |
| Extrusion along custom path | + | − | + | − | + | − | − | + |

**Table 2.1:** Overview and comparison of related projects regarding the MR continuum (− = Desktop application, not part of MR continuum) and categories (− = not supported or integrated, + = supported).

### Evaluation

Projects such as the Mockup Builder [17] and Construct3D [18] try to enhance direct user interaction with immersive and semi-immersive technologies. By using HMDs, designing tasks can be supported with additional stereoscopic visualizations. Mockup Builder even creates midair experiences since they work with finger tracking and depth cameras [17]. Therefore, Mockup Builder has got a very positive impact on usability because interaction seems to be more natural, but a disadvantage of midair interaction is the lack of accuracy and precision [8]. However, it seems that there is a lack of evaluating the performance of 2D modeling with planar visualization mapped to stereoscopic visualization compared to conventional 3D modeling systems with which users can directly work in 3D.

## 2.4 Comparison and Conclusion

In the table 2.1, a few projects are listed and compared in terms of the MR continuum and how they can be categorized.

The projects are also analyzed in terms of stereoscopic visualization and usage of at least one HMD and if the system can be used in collaboration with multiple users.

Additionally, the listed projects are also compared by their input devices, if a PIP is integrated, if freehand drawing is possible and if there are also shapes, patterns, and gestures predefined to use out of the box, and finally if the application supports extrusion in any form or extrusion along a custom path in particular. The mentioned projects [12, 17, 18] can be tagged as an AR application since they provide all characteristics of an AR system that have been referred to in section 2.1.1. Amongst others, they also use at least one HMD for presenting virtual objects as a stereoscopic visualization to the user at which Mockup Builder [17] does not support multiple HMD devices for collaborative working.

As shown in table 2.1, not all of the included projects use a pen as their primary input device but also the mouse (especially the Desktop applications), touch input, finger tracking or even provide a PIP to the user to directly interact with an augmented menu. Almost all of the listed projects offer freehand drawing except Construct3D [18] which merely offers the creation of predefined 2D and 3D shapes. ILoveSketch [3] mainly supports predefined gestures and patterns for user interaction which are very similar to traditional paper and pen drawing and SKETCH [32] focuses on a purely gestural interface to be as natural as pen-and-paper sketching.

Extrusion is divided into general extrusion, which summarizes various forms of extrusion and extrusion along a custom path. General extrusion refers to transforming 2D shapes into 3D objects by any extrusion algorithm, no matter if it is incremental extrusion, extrusion along a straight path or rather along an axis or creating 3D objects by rotating or sweeping a 2D shape.

Extrusion along a custom path has been carved out because it is relevant in particular for the master thesis and is featured in SESAME [24], Mockup Builder [17], Teddy [16] and SKETCH [32] whereas Construct3D [18] and Studierstube [12] do not support extrusion. ConMan [14] realizes extrusion by rotating around an axis and sweeping 2D shapes along an axis. ILoveSketch [3] only provides extrusion along an axis.

# Chapter 3

# Conceptual Design

This chapter describes the development process of the conceptual design of the prototype in detail. It contains the initial goals as well as the system requirements which are demanded from the prototype, the system design, a big picture of the system and user interface and interaction design sketches.

## 3.1 Initial Goals

The role of the thesis project is not to implement a modeling tool, but to focus on extrusion, amongst the set of possible modeling operations. The main goal is to implement an input modality for a 2D interface, which enables users to draw custom 2D shapes and to extrude them into 3D space. The extruded 3D objects should be visualized either planar on the surface or as a stereoscopic presentation on an HMD.

Regarding the motivation for the thesis project, specific goals have been set up to realize the idea of such a system. The requirements are defined in the section 3.2.

## 3.2 System Requirements

Requirements for the project were specified in order to define the scope of the project and to clarify the necessary steps for the realization of the project idea. The requirements are introduced in the following and give a good overview of various features the project should contain:

- communication between HMD and tablet,
- drawing of custom 2D shapes,
- drawing of custom extrusion paths,
- extrusion,
- stereoscopic visualization,
- marker recognition and tracking and
- pen interaction.

### 3.2.1 Communication between HMD and Tablet

The communication between the tablet-based interface (*Microsoft Surface*) and the see-through HMD (*Microsoft HoloLens*) is an essential feature for the realization of this project because the surface must send geometry data, which is needed by the HMD to create the extruded object. Therefore, communication over a network connection is set as a requirement for the project.

### 3.2.2 Drawing Custom 2D Shapes and Extrusion Path

Another feature is to create a canvas on which the user can draw custom line segments to form a custom 2D shape and the extrusion path. It would also be desirable to detect closed shapes.

### 3.2.3 Extrusion

The most important feature for the project is the ability to extrude a previously drawn 2D shape into 3D space by drawing an extrusion path. It should be possible for the user to switch the mode from drawing the custom 2D shape to drawing the extrusion path. This switch should be visualized by assigning colors to the lines respectively to the active mode.

### 3.2.4 Planar and Stereoscopic Visualization of 3D Objects

The created 3D objects should be visualized either planar on a surface or stereoscopic. The stereoscopic visualization will be presented with the help of an HMD and should be placed either directly on the tablet-based surface with the help of a tracked image or beside the surface. This requirement is essential for the user study and the evaluation later on in the thesis.

### 3.2.5 Marker Recognition and Tracking

It should be possible to recognize and track a planar image (marker) with the *Microsoft HoloLens* because it is necessary to be able to place a newly created and extruded 3D objects on the tablet-based surface. Recognition and tracking of markers should be realized with the Augmented Reality SDK Vuforia.

### 3.2.6 Pen Interaction

The user should be able to interact and draw the custom 2D shape and the extrusion path by using a pen that belongs to the chosen *Microsoft Surface.*

## 3.3 System Design

This section describes the project in a more detailed way. For instance, system architecture and design aspects will be described.

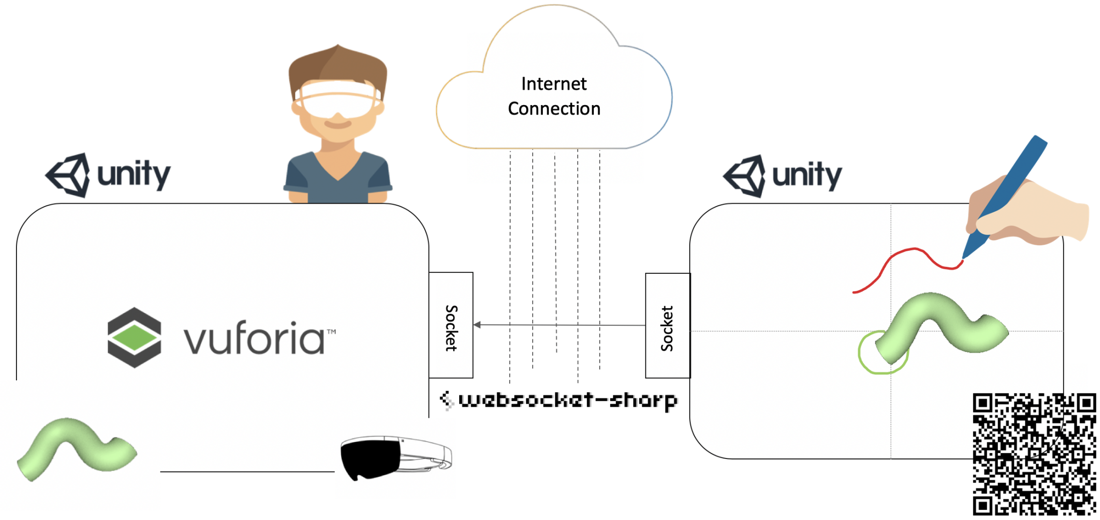


**Figure 3.1:** Overview of the System Architecture

### 3.3.1 Big Picture

As shown in figure 3.1, the thesis project consists of two applications: one for the tablet-based surface and one for the HMD.

The user holds the tablet-based surface in his hands and draws a custom 2D shape. The custom 2D shape is indicated as a green line in the figure 3.1 and is drawn on a planar surface. By drawing another line path, the user can define the extrusion path of the 2D shape. The extrusion path is indicated as a red line in the figure 3.1. Accordingly, the 2D shape will extrude into 3D space along the user-defined extrusion path. The extruded 3D object is presented as a green, three-dimensional pipe in the figure 3.1. The 3D object can be displayed either on the planar surface (right side) or as an AR-based stereoscopic visualization (left side with HMD) directly above or alongside the 2D surface. The stereoscopic visualization is placed respectively to a planar image (Vuforia *Image Target*) with which Vuforia can recognize and track the tablet-based surface and its position. Additionally, the user can apply fundamental transformations such as translation, rotation, and scaling and can select and delete objects. Both applications are communicating with each other over a network connection. The applications use Websockets for communication.

## 3.4 User Interface and Interaction

The following section describes the process of developing the user interface and also the user interaction with previously created objects. The major design decisions will be explained.

### 3.4.1 Interface Design

The figure 3.2 presents four major steps of the design process of the tablet user interface.

### Initial Iteration

The first sketch in the left upper corner is the first sketch and focuses on a straightforward and intuitive interface. It only includes three buttons for extruding, rotating and translating the extruded 3D object. The sketch also contains a coordinate system to indicate the plane in which the user is operating. Similar to other modeling tools, the x-axis has the color red, the y-axis is always marked with the color green, and the z-axis is blue. At the top of the screen, three additional buttons show the user the three possible dimensions. For example, if the button z is chosen, the 2D shape will be drawn in the x and y plane, and the shape will be extruded in z-direction. Otherwise, if x is chosen, the user extrudes along the x-axis.

An advantage of this approach would be that the user may work more accurately because when changing the dimension with one of the three buttons on top, the user views the object from a different perspective and can draw the shape relative to the present object.

Furthermore, it also becomes possible to extrude in all directions with one extrusion path. In this case, the user would draw the custom 2D shape, for example, in the x and y plane. Then he selects the z button and starts with drawing the extrusion path or extruding the shape in the z-direction. Then he potentially stops with drawing and selects x to continue extruding in the x-direction.

Because of the fixed coordinate system in the current prototype, which is always set to the xy-plane as the drawing canvas on the tablet, extrusion is limited to the yz-plane.

### Second Iteration

As illustrated in the second sketch in the upper right corner in figure 3.2, the three buttons on top can be discarded and replaced by rotation of the object. The extrusion along all three axes is not possible in this approach and is limited to two directions.

The second sketch is closer related to a standard modeling tool since the buttons from the first sketch have been extended with further operations. The buttons are now positioned at the top border to avoid unintended button events and include object selection (hand icon), translation, rotation, scaling, extrusion of shape, drawing a shape, undo and object deletion.

Compared to the first sketch, extruded 3D objects are not visible within the drawing canvas in the second sketch. Therefore, a preview window has been added in the lower right corner to offer a planar visualization of the extruded objects to the user and to allow editing in 3D space. The idea is to apply object manipulation on the shape, which represents the related 3D object in 2D and to view the manipulation in 3D space in the preview window.

### Third Iteration

The third sketch is closely related to the second sketch. An add button has been added for creating multiple objects by adding new ones. It also considers the color coding of the lines. Previously, the shape was red, and the extrusion path was blue, which can be confusing when thinking of axis colors. The third sketch also demonstrates how a translation of an extruded object can look like, which is described in more detail in

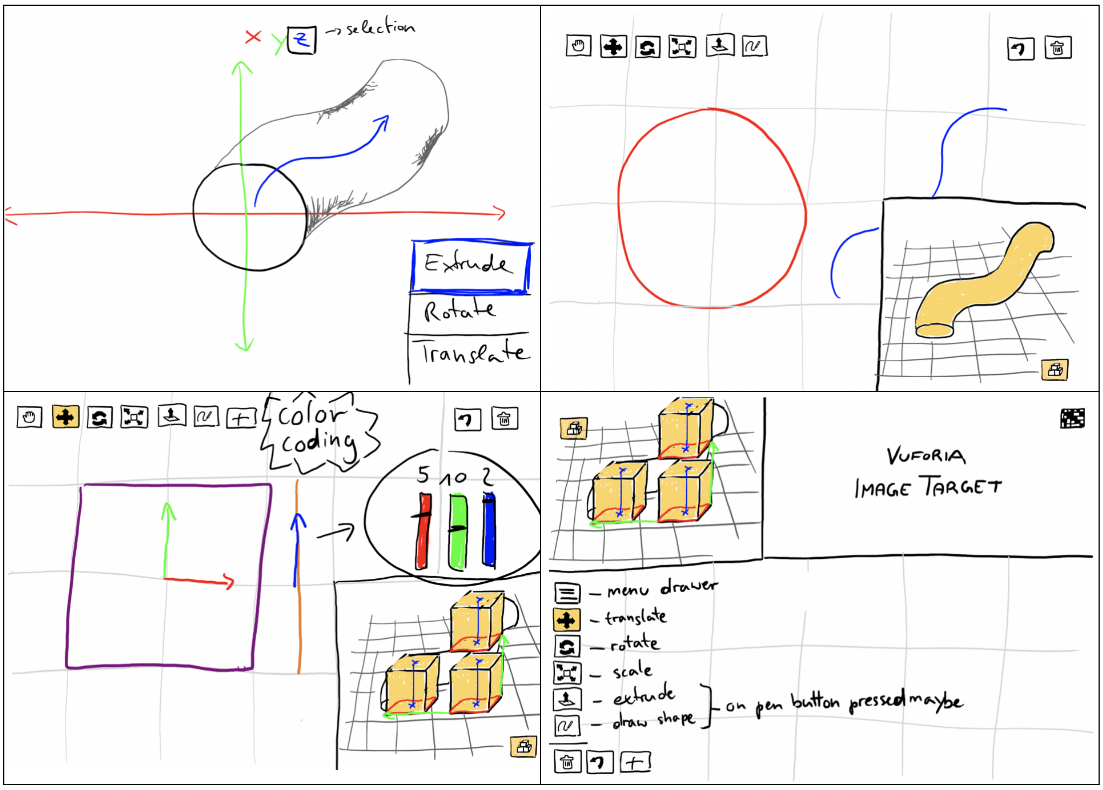


**Figure 3.2:** Iterations in the design process for the user interface of the tablet application.

section 3.4.2. The cube and the translation of it can be viewed in the preview window. The object is selected if it is colored yellow. The corresponding shape and extrusion path are then displayed on the drawing canvas. Hence if no object is selected, the canvas would be empty.

### Fourth Iteration

In the fourth sketch in the lower right corner of figure 3.2, the major components, which are the buttons, the drawing canvas, and the preview window, have been repositioned.

The preview window is now placed in the upper left corner to avoid occluding the preview window with the own hand. At this point, a design decision had to be made to focus the layout on right-handed people. The buttons have moved to the left border of the screen to make room for the tracker at the top and to have a uniform shape of the drawing canvas (rectangle) instead of a patchwork surface with more corners which may irritate the user during the drawing process. The buttons have been extended with a menu button which will serve as an entry point for the file management system to save and load a scene. A marker was also added for tracking the tablet and for registering and placing the stereoscopic visualization of the extrusion object. If the marker is not required, because the user wants to rely on the planar visualization in the preview window, then he can hide the marker by pressing the marker button in the upper right corner. By pressing it again, he can open the marker again.

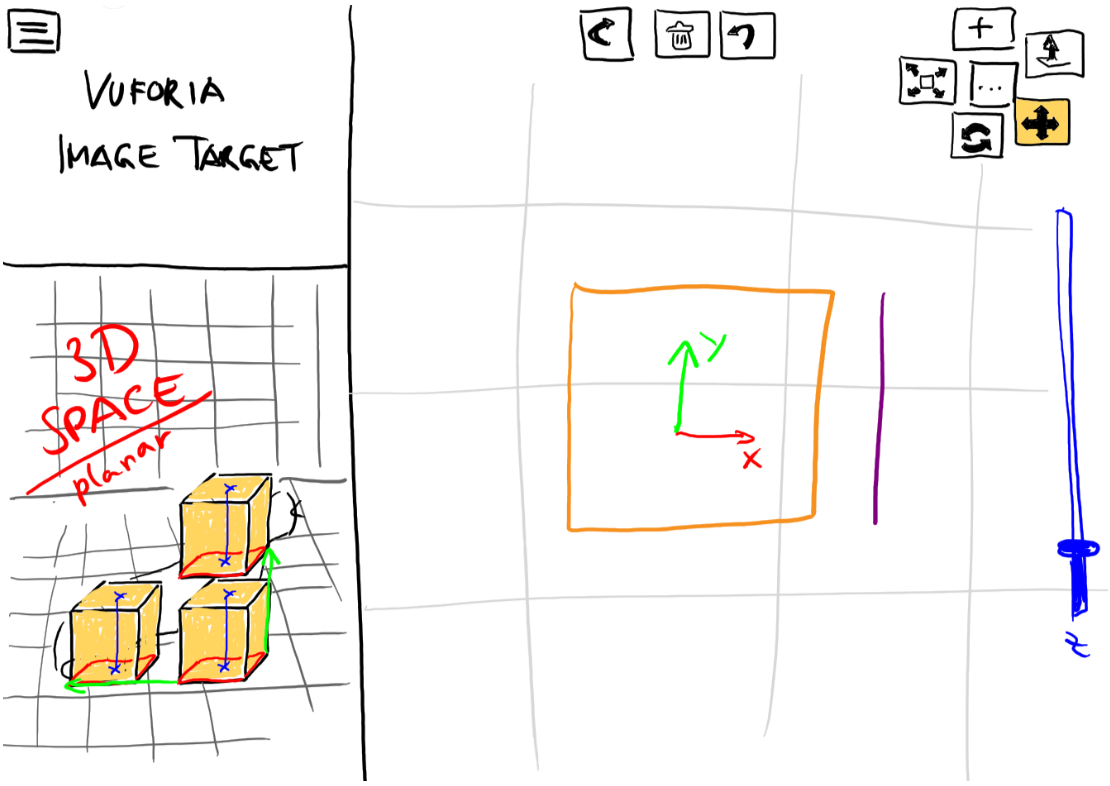


**Figure 3.3:** Final version of the tablet user interface.

### Final Iteration

The final result of the user interface is shown in figure 3.3. As illustrated in the sketch, the components have been repositioned again to receive a more quadratic drawing canvas. Another advantage of this arrangement is the bigger preview window, which offers a better overview of the scene.

Furthermore, the marker is placed in the upper left corner which is the place best visible to the HMD[1] provided that the user is right-handed.

The buttons can now be hidden by pressing on the dots button in order to save space. When pressing again, the buttons will appear again. The undo, redo, and delete buttons are now placed at the top of the screen, which feels unobtrusive as well. The text describing the functionality of each button has also disappeared, and appropriate button icons have been chosen for each functionality.

## 3.4.2 Interaction Design

This section describes different ways of how the user interaction regarding selection and the basic object transformations could be realized in the prototype.

[1] The marker should always be visible to achieve a stable tracking of the marker.

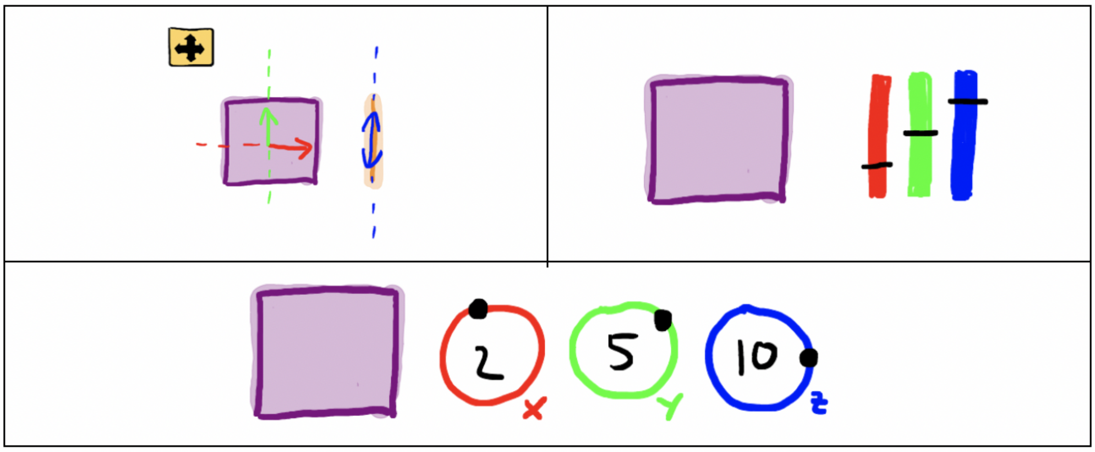


**Figure 3.4:** Three sketched opportunities for object translation.

### Selection

Selection is an essential part of the manipulation of objects in three-dimensional space. In order to manipulate an object with the typical object transformations, the desired object must be selected first. According to Bowman et al. [6], there are different options to select an object in the context of a virtual environment. As it is in reality, humans are selecting an object with their hands. The equivalent in the virtual environment is giving the user a virtual hand in the form of a 3D cursor. Concerning the prototype, however, this approach is not an option as the input device should only be the pen. A mixing of interactions with the pen and hand gestures for selecting objects with the 3D cursor of the HMD should be avoided.

In the case of the prototype, a suitable option for selecting objects would be the ray-casting approach. The user is pointing at an object, and a ray is cast into the scene. If the ray hits the object, it will be selected. Instead of pointing with a 3D cursor, as described by Bowman et al. [6], the user is using the pen to touch the object on the tablet, and a ray is cast into the scene. Hence, a mixture of input modalities is prevented, and the user can work solely with the pen.

### Translation

Figure 3.4 shows three sketches which present possible realizations of object translation. The first sketch includes a translation gizmo (similar to modeling tools) which is placed in the center point of the shape. The user can translate it by clicking the gizmo handles and translating the shape. The translation handle for the z-direction is placed on the extrusion path because extrusion is always along the z-axis and it looks more uniform than a z-translation slider. Nevertheless, a drawback of the translation gizmo is the overlap of two lines. For example, if the user is translating the square over the extrusion path, the extrusion path may not be visible or clickable anymore, and translation in z-direction would be blocked. The sketch in figure 3.5 may be a solution to this problem. By using the pen button, the dimension is switching from xy to xz, which means that

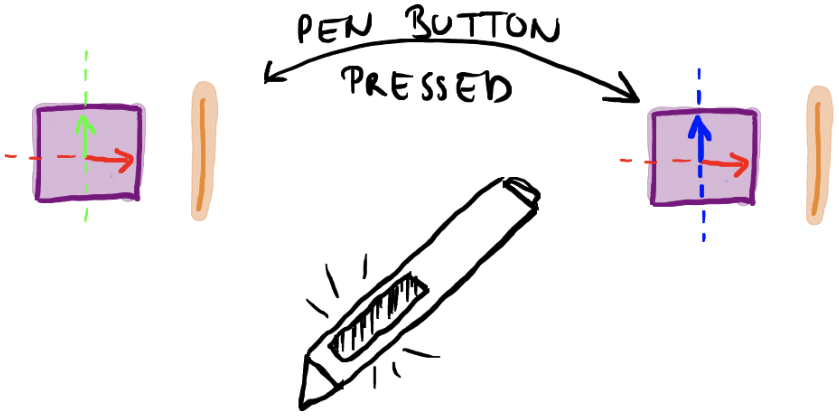


**Figure 3.5:** Translation with pen interaction.

a translation in z-direction would be possible. Nevertheless, using a pen button is not intuitive and evident to the user and switching the translation dimension is also more complex to visualize because the square in the xy-plane should be presented as a line in the xz-plane.

The second sketch in figure 3.4 uses only sliders to provide translation in all directions. The main disadvantage of these sliders is that they have a limited range. By using circular sliders with an infinite range as illustrated in the third sketch, this disadvantage could be overcome.

### Rotation

In figure 3.6, three ways to rotate an object are illustrated. The first sketch is equivalent to the first translation sketch in figure 3.4 and uses a rotation gizmo for indicating the three rotation handles. Compared to the translation gizmo, all three handles are available simultaneously.

The second sketch is based on a paper by Stürzlinger et al. [9] about a one-handed multi-touch method for 3D rotations. When touching the object with a finger and by moving it, the object will be rotated in the corresponding direction, which is either in x- or y-direction. When holding down one finger, the user can rotate the object in the z-direction by using a second finger.

An advantage is that this version of rotation is also very compact, similar to the first rotation sketch since there are no additional UI elements needed such as sliders. However, the main disadvantage is to break the input modality by using touch input instead of the pen input. The user would have to switch frequently between finger and pen interaction.

The third rotation sketch is equivalent to the third translation sketch and uses infinite sliders for object rotation.

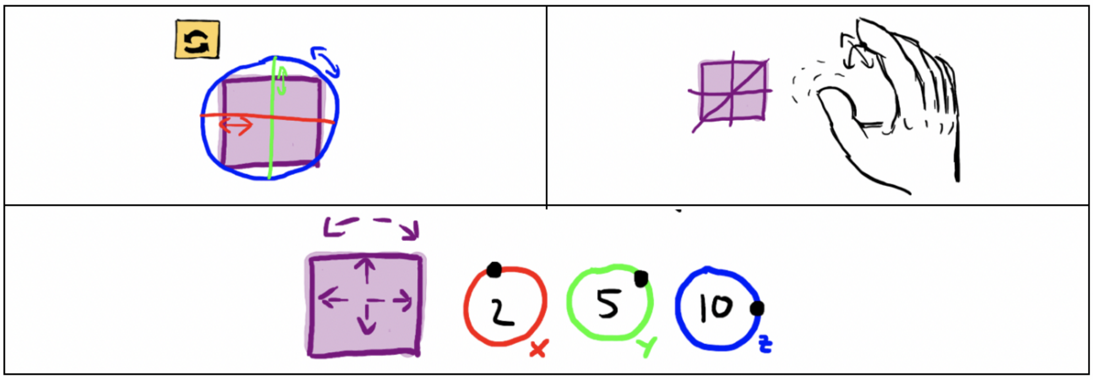


**Figure 3.6:** Three sketched opportunities for object rotation.

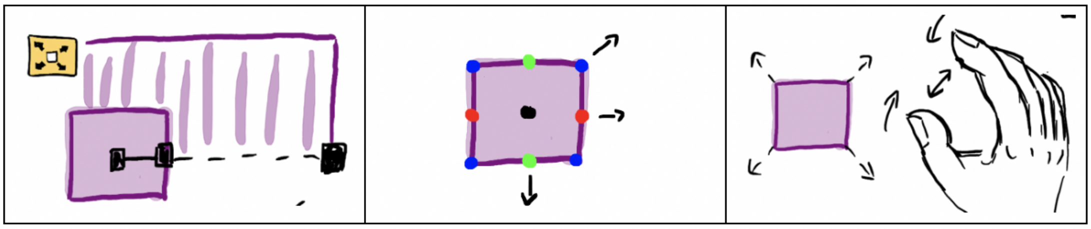


**Figure 3.7:** Three sketched opportunities for object scaling.

### Scaling

In figure 3.7, three scaling sketches are presented. The first sketch shows uniform scaling, which means that the user can scale the whole object homogeneously in all directions by pressing the black dot in the center of the shape and drag it. Nevertheless, this approach limits the user and scaling the object in only one single direction is not possible.

The second sketch extends the first sketch. The black dot in the center of the shape is for uniform scaling. The red, green, and blue dots represent the three axes. For example, by clicking on a red dot, the user can scale the object along the, whereas clicking on a blue dot and dragging it will scale the object along the z-axis.

The third sketch is using touch input for scaling. By doing the pinch to zoom gesture, the object will be homogeneously scaled in all directions simultaneously. As already mentioned in the rotation section, using touch input breaks the input modality, which has been chosen for the prototype, which uses only pen input. Nevertheless, the zooming gesture is a very intuitive gesture for scaling.

## 3.4.3 Placement of AR Marker

Different options exist for placing a marker[2] on a surface. For the prototype, a *Vuforia Image Target* was used for tracking the tablet and for registering the stereoscopic visualization on the surface.

[2] A marker is a planar and trackable image.

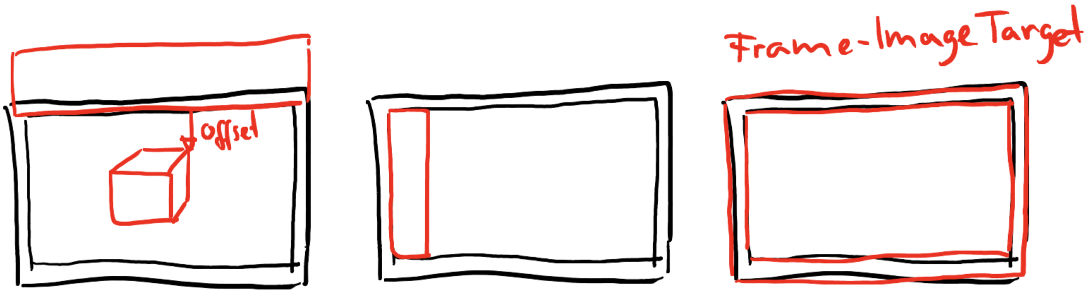


**Figure 3.8:** Three opportunities for placing the Image Target on the planar surface.

In figure 3.8, three ways for placing a marker on a surface are illustrated. One option is to place the marker at the top border of the tablet and add an offset to the virtual objects so that they appear in the correct position. This option would save space because the marker does not need to be presented in the user interface. Hence, additionally saved space can be spent on other components in the user interface.

The main drawback of this approach is the need for a printed marker which requires the user to always carry a sheet with him in order to be able to see the stereoscopic visualization.

The marker in the second sketch is placed directly in the user interface at the left border. Since the user interface is adapted for right-handed users, the marker on the left side will be visible for the HMD most of the time. The disadvantage of this option is the wasting space marker. The size of the drawing canvas automatically decreases by adding the marker to the user interface, but at least the user does not need an additional sheet with a printed marker.

The third sketch illustrates a marker which has the form of a frame. It is placed around the user interface. Hence it does not waste space and carrying a printed marker is not necessary if the marker is attached permanently on the tablet.

Unfortunately, this approach is not feasible so far because of the technical limits of the *Vuforia* library. *Vuforia* does not support markers with the shape of a frame.

Furthermore, Vuforia Model Targets have also been taken into account but have been discarded because a tablet does not offer enough adequate characteristics to serve as a Model Target. For instance, a Model Target has to feature stable surface features, and unfortunately, shiny surfaces are not supported.

### 3.4.4 Network Communication

A network connection between the HMD and the tablet is required to send data between both opponents.

Hence, a traditional server and client architecture could be used to send data between both applications, whereas the tablet will serve as the server, and the HMD will connect to the server as a client. Both applications will run the extrusion algorithm, and the server only sends user input data or rather the mouse position of the pen to keep the communication protocol simple.

# Chapter 4

# Technical Implementation

This chapter presents the whole prototype system, consisting of the tablet and HMD application, which has been developed in conjunction with this master thesis. It describes the technical implementation of the prototype which includes the technology stack used for the prototype, the system architecture, multiple system components as well as the build and deployment process for the HMD and tablet application.

## 4.1 Software

The following technology stack has been chosen for the realization of the prototype. The technology stack is an assembly of:

- *Unity3D*[1],
- *Websocket-sharp*[2] and
- *Vuforia Augmented Reality SDK*[3].

### 4.1.1 *Unity3D*

*Unity3D* gives users the ability to create games in both 2D and 3D, and the engine offers a scripting API in C sharp. *Unity3D* supports the creation of custom vertices, fragments, and procedural mesh generation, which is required for the prototype since extrusion of a 2D shape is part of the application.

*Unity3D* serves as the development environment for both applications and is a cross-platform game engine that supports the needed platforms. The application for the *Microsoft HoloLens* (used HMD for this prototype) is built and deployed as a *Universal Windows Platform* (UWP) application and the *Microsoft Surface* (used tablet-based surface for this prototype) application is built and deployed as a *Windows Standalone* application.

[1] Unity 3D is a cross-platform game engine for creating 2D and 3D applications.

[2] Websocket-sharp is used for providing a Server and Client communication structure.

[3] The Vuforia SDK provides recognition and tracking of planar images.

### 4.1.2 *Websocket-sharp*

*Websocket-sharp* is a library that provides the `WebSocket` protocol client and server for the communication between the HoloLens and the Surface. Because *Websocket-sharp* does not support UWP applications, a wrapper class for the HoloLens application has been integrated to enable the usage of *Websocket-sharp*. Underneath, the application is still using UWP sockets. *Websocket-sharp* also supports secure connections, HTTP authentication, and connections through the HTTP Proxy server.

### 4.1.3 *Vuforia Augmented Reality SDK*

*Vuforia* is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision algorithms to recognize and track planar images (*Image Targets*) and simple 3D objects (*Model Targets*) in real-time. This image registration capability enables developers to position and orient virtual objects, such as 3D objects, with real-world images when they are viewed through the camera of a mobile device.

## 4.2 Hardware

For the concrete implementation of the prototype, the following hardware components were chosen:

- *Microsoft Surface*[4],
- *Microsoft Surface Pen* and the
- *Microsoft HoloLens*[5].

### 4.2.1 *Microsoft Surface*

The *Microsoft Surface* is a hybrid of a touchscreen interface and a personal computer and provides the option of attaching and detaching a keyboard. The surface can also be used with a digital pen as a touch input device. The Surface family features multiple lines of devices such as the *Surface Laptop* or the *Surface Pro*.

For the related prototype, the *Surface 3* is used in combination with the *Microsoft Surface Pen*. The prototype application for the surface is built as a *Windows Standalone* application.

### 4.2.2 *Microsoft Surface Pen*

The *Surface Pen* is a digital pen used as an optional input device for Surface devices. The pen features a button on the side which simulates the right-click with a mouse. There is also an eraser button on top of the pen, which deletes pen strokes when touching the tablet.

[4] The *Microsoft Surface* serves as the tablet-based interface for the user to interact with and draw the custom 2D shape and the extrusion path.

[5] The *Microsoft HoloLens* serves as see-through HMD to display stereoscopic visualization in the field of view of the user.

Unfortunately, Unity does not support pen input, and therefore, pen buttons cannot be used in the prototype because the pen input in Unity is treated similar to a standard finger touch input.

### 4.2.3 *Microsoft HoloLens*

The *Microsoft HoloLens* is a see-through HMD and features different sensors and hardware components to enable developers to create realistic and natural AR experiences.

For example, the HoloLens has its processors such as CPU, GPU, and the *Microsoft Holographic Processing Unit* (HPU). It also includes an inertial measurement unit (IMU) that incorporates sensors such as an accelerometer, a gyroscope, and a magnetometer. With the HPU, the HoloLens can fetch data from all included sensors and process them for further features such as spatial mapping, gesture recognition as well as speech recognition.

The HoloLens also contains a depth camera, a video camera, a microphone, audio speakers as well as a light sensor and also owns projection lenses for presenting the images to the user. Unfortunately, the field of view of the HoloLens is currently limited to approximately 30 degrees.

The device is running the *Windows 10* operating system and applications are built for the *Universal Windows Platform* (UWP).

## 4.3 System Architecture of the Tablet Application

The architecture of the Surface application is illustrated in the figure 4.1. The figure includes the major classes of the prototype and how they are related to each other. In the following section, system components such as the network connection, extrusion algorithm, and planar visualization of extruded objects will be described in detail.

### 4.3.1 Network Connection and Communication

The *websocket-sharp* library has been used to build a classic server and client structure in order to let the two applications communicate with each other. Figure 4.2 (a) shows how the `WebSocket` server is created in the class `MyServer` and started on the *Microsoft Surface* whereas figure 4.2 (b) presents how the `WebSocket` client is created in the class `WebSocketClient` and started on the *Microsoft HoloLens*.

#### UWP Wrapper

Unfortunately, the *websocket-sharp* library does not support the *Universal Windows Platform* (UWP) which is the platform for development on the *Microsoft HoloLens*. Therefore, a wrapper class is necessary, which replaces the `WebSocket` class of the library if the application is built for the UWP platform. Hence, when the application is executed in the Unity Editor, the *websocket-sharp* library is used, but if the application is built as a UWP app and executed, the application will use for instance the `Windows.Networking.Sockets.MessageWebSocket` class instead of the `WebSocket` class of the *websocket-sharp* library.

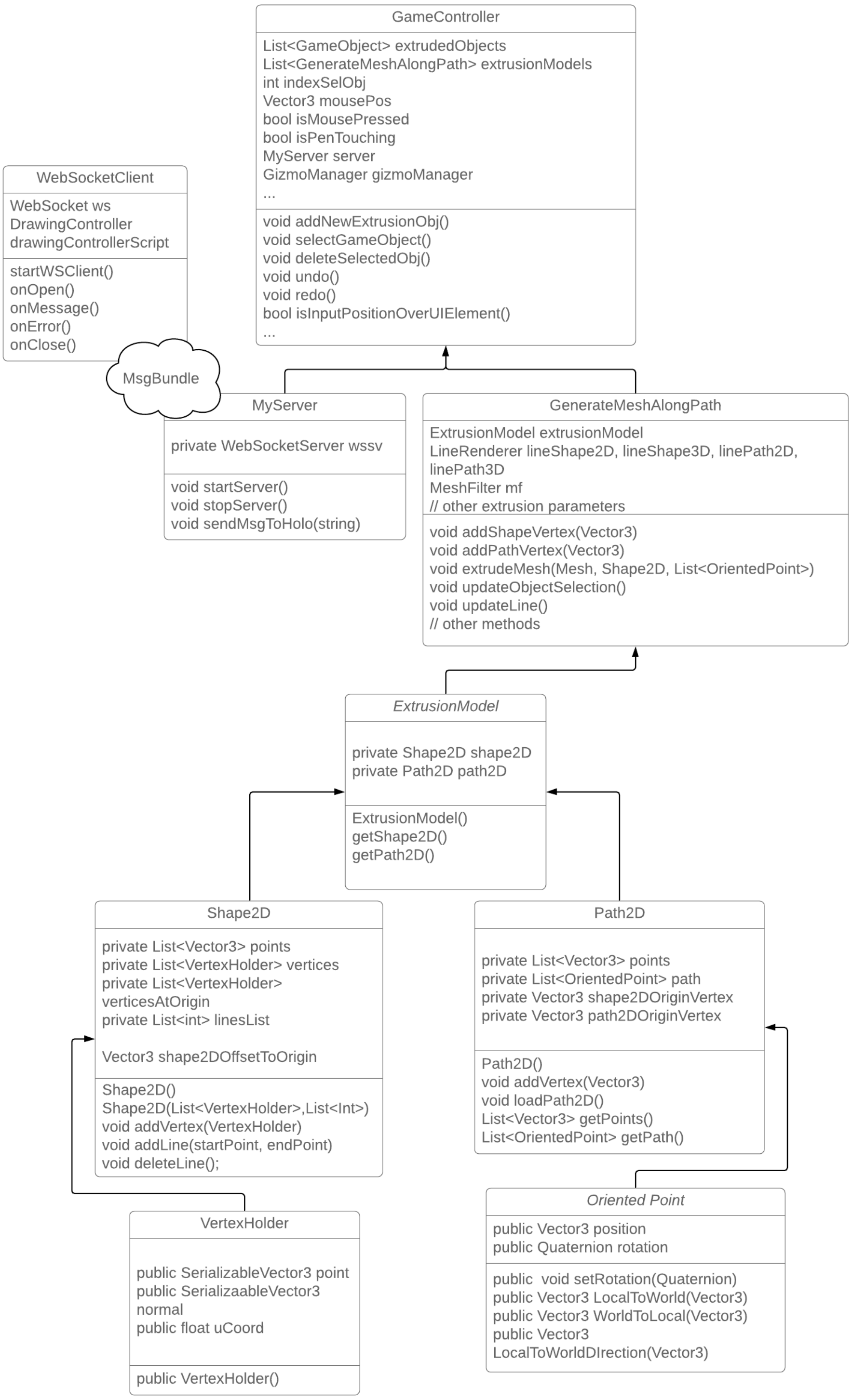
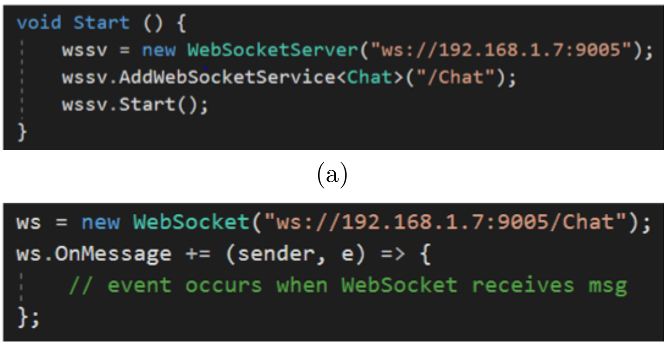

GameController
List<GameObject> extrudedObjects
List<GenerateMeshAlongPath> extrusionModels
int indexSelObj
Vector3 mousePos
bool isMousePressed
bool isPenTouching
MyServer server
GizmoManager gizmoManager
...
void addNewExtrusionObj()
void selectGameObject()
void deleteSelectedObj()
void undo()
void redo()
bool isInputPositionOverUIElement()
...
WebSocketClient
WebSocket ws
DrawingController
drawingControllerScript
startWSClient()
onOpen()
onMessage()
onError()
onClose()
MsgBundle
MyServer
private WebSocketServer wssv
void startServer()
void stopServer()
void sendMsgToHolo(string)
GenerateMeshAlongPath
ExtrusionModel extrusionModel
LineRenderer lineShape2D, lineShape3D, linePath2D,
linePath3D
MeshFilter mf
// other extrusion parameters
void addShapeVertex(Vector3)
void addPathVertex(Vector3)
void extrudeMesh(Mesh, Shape2D, List<OrientedPoint>)
void updateObjectSelection()
void updateLine()
// other methods
ExtrusionModel
private Shape2D shape2D
private Path2D path2D
ExtrusionModel()
getShape2D()
getPath2D()
Shape2D
private List<Vector3> points
private List<VertexHolder> vertices
private List<VertexHolder>
verticesAtOrigin
private List<int> linesList
Vector3 shape2DOffsetToOrigin
Shape2D()
Shape2D(List<VertexHolder>,List<Int>)
void addVertex(VertexHolder)
void addLine(startPoint, endPoint)
void deleteLine();
Path2D
private List<Vector3> points
private List<OrientedPoint> path
private Vector3 shape2DOriginVertex
private Vector3 path2DOriginVertex
Path2D()
void addVertex(Vector3)
void loadPath2D()
List<Vector3> getPoints()
List<OrientedPoint> getPath()
VertexHolder
public SerializableVector3 point
public SerializaableVector3
normal
public float uCoord
public VertexHolder()
Oriented Point
public Vector3 position
public Quaternion rotation
public  void setRotation(Quaternion)
public Vector3 LocalToWorld(Vector3)
public Vector3 WorldToLocal(Vector3)
public Vector3
LocalToWorldDIrection(Vector3)

**Figure 4.1:** Class diagram representing the system architecture and the relations of the main components.



```
void Start () {
    wssv = new WebSocketServer("ws://192.168.1.7:9005");
    wssv.AddWebSocketService<Chat>("/Chat");
    wssv.Start();
}
```

(a)

```
ws = new WebSocket("ws://192.168.1.7:9005/Chat");
ws.OnMessage += (sender, e) => {
    // event occurs when WebSocket receives msg
};
```

(b)

**Figure 4.2:** Server-side connection on the *Microsoft Surface* (a) and client-side connection on the *Microsoft HoloLens* (b).

`MyServer`

When the application starts, an instance of `MyServer` is generated, and the server starts and listens on a port for incoming messages of type `MsgBundle` which is illustrated in figure 4.1. The `WebSocketClient` class belongs to the HoloLens application and communicates with the `MyServer` class over a WLAN connection in the Surface application.

`MsgBundle`

As shown in figure 4.1, the `MsgBundle` class is used to send messages between the surface and the HoloLens in a consistent manner. The `MsgBundle` contains fields to store the current editor mode code, which is defined in the `GameSettings` class. The `MsgBundle` class also includes another integer field as well as multiple vector fields for transmitting, for example, transform data of an object.

### 4.3.2 UI Controller

The `GameController` class, presented in figure 4.1, is an essential class in which multiple strings are pulled. For example, the `GameController` holds a list of game objects which represent the created extrusion objects by the user. These extrusion objects are instantiated from a prefab in the *Resources* folder in Unity, which is called ExtrusionObj. Each instantiated `ExtrusionObj` contains a script component called `GenereateMeshAlongPath` which are also collected into a list similar to the objects list.

The `GameController` also contains the field `indexSelObj` which keeps track of the currently selected extrusion object. The value of `indexSelObj` indicates the index of the selected object in the `extrudedObjects` list and the corresponding script component in the `extrusionModels` list.

Furthermore, this class is the entry point for input events. If the user is touching the surface and triggers an input event, the `GameController` will process the input position of the touch input. For instance, the `GameController` class checks if the current input position is inside the preview window or the drawing canvas and will process the position accordingly. If the input happened to be inside the preview window, a ray is cast into the scene, and a new object may be selected (see section 4.3.5). Otherwise, the input position will be used as a vertex of a shape or a path line.

If the user presses buttons, for example, the add, delete, undo or redo button, the appropriate method will be called in the `GameController`.

### 4.3.3 Extrusion

The `ExtrusionModel` is not a `MonoBehaviour` class and is used as a data holder. It contains a `Shape2D` and a `Path2D` object.

The `Shape2D` class describes the data for a custom 2D shape and contains its vertices. When a new vertex is added, the vertex is also stored in the `verticesAtOrigin` list. Each vertex is subtracted from the first vertex (`shape2DOffsetToOrigin`) to translate the whole shape back to the origin. The extrusion algorithm only uses `verticesAtOrigin`. During extrusion, it is necessary to calculate the mesh with the shape being translated to the origin.

The `Path2D` class describes the data for a custom 2D extrusion path and contains its vertices. Every time a new vertex is added to the path, the vertex is used for calculating the corresponding `OrientedPoint`, which will be added to the list of `OrientedPoints`.

The extruded object depends on the custom 2D shape the user is drawing at first (in draw shape mode) and also depends on the 2D extrusion path the user is drawing afterwards (in extrusion mode) on the drawing canvas.

#### Custom 2D Shape

As presented in figure 4.3, the custom 2D shape is marked as a yellow line. The points of the custom 2D shape are stored in a list. Additionally, the lines between two points of the custom 2D shape are also stored in a separate list. A start and endpoint define each line. The start and endpoints are referenced by using the corresponding vertex indices in the points list. So, each custom 2D shape contains a list of points and a list of lines.

#### Custom Extrusion Path

As shown in figure 4.3, the custom extrusion path is marked with red dots. As with the custom 2D shape, the points of the extrusion path are also stored in a list. Additionally, an orientation point must be derived for each point in the list, because the newly generated vertices must be rotated perpendicular along the extrusion path. Not rotating the newly generated vertices will result in an orientation problem, which is shown in the left image in figure 4.4. The extrusion path is a wave-formed path and the cross-sections, which contain the new vertices, all have the same orientation, but this is not a suspected outcome of extrusion along a path. Hence, rotation must be calculated for each cross-section, so that the cross-section is perpendicular to the extrusion path as it is in the right image in figure 4.4.

An oriented point contains a position that matches the vertex of the path as well as a rotation which is the facing direction or rather the direction, and each shape vertex must be rotated with to be perpendicular to the extrusion path. As presented in figure 4.3, the

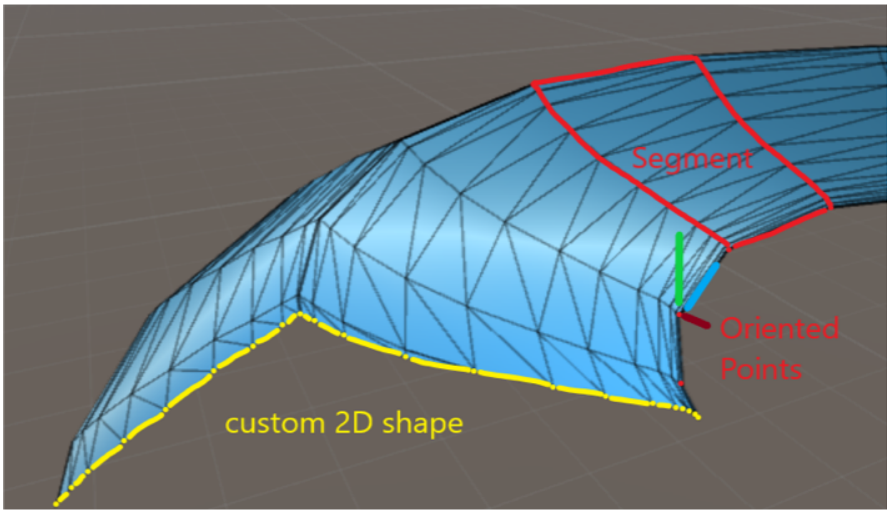


**Figure 4.3:** Custom 2D shape (yellow), extrusion path with its `Oriented Points` (red dots) and segments with its triangles.

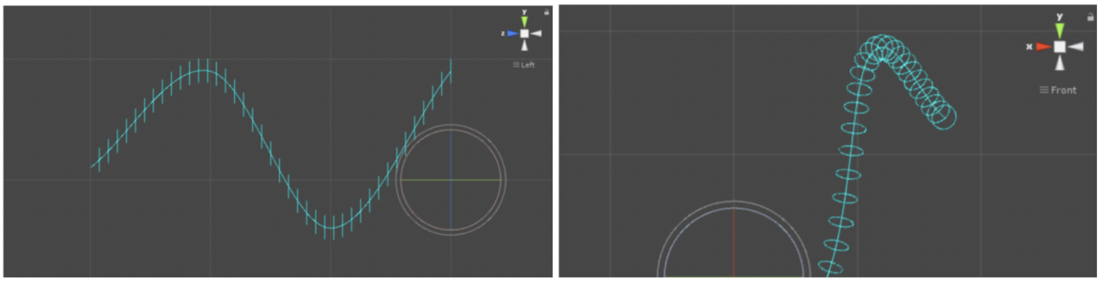


**Figure 4.4:** Incorrect rotation of cross-sections (circles) along the extrusion path.

tangent (blue), the binormal (red), and the normal (green) are necessary for calculating the facing direction at each path vertex.

**Algorithm 4.1:** Derivation of an Oriented Point

```
prev = previous path point
curr = current path point
tangent = curr - prev
binormal = cross product(up, tangent)                    ▷ up vector = (0, 1, 0)
normal = cross product(tangent, binormal)
facing direction = normal
add oriented point(position = curr, rotation = facing direction)
```

The algorithm 4.1 shows how the facing direction of the newly added vertices is calculated. The vertices are copied from the original shape and rotated perpendicular to and translated along the extrusion path.

The first step is to calculate the tangent vector out of the previous path point and the current path point. The tangent results from subtracting the previous point from the current and therefore lastly added path point.

Next, the binormal is calculated with the cross product of the up vector (which is $(0, 1, 0)$ in Unity) and the tangent. The cross product of two vectors always delivers the perpendicular vector. Afterwards, the normal vector is calculated with the cross product tangent and binormal. The normal is always perpendicular to the path point and is used as the facing direction.

Finally, to generate the mesh, segments must be added as illustrated in figure 4.3. A segment contains multiple triangles that form the mesh.

### Procedural Mesh Generation

Since extruded objects consist of a mesh, and the mesh is dependent on the user input, the mesh needs to be generated dynamically in code at runtime. Creating meshes dynamically at runtime is called procedural mesh generation.

In algorithm 4.2, the pseudo-code for the extrusion process is presented. First of all, iterating over each oriented point in the extrusion path is necessary to extrude to the last drawn point. For each oriented point, an inner loop is started which iterates over all points of the custom 2D shape. For each vertex of the 2D shape, a new vertex is created with its position relative to the oriented point. Also, a normal direction is added. Now that we have all new vertices created, it is necessary also to connect them with triangles to generate a mesh.

**Algorithm 4.2:** Extrusion

```
for each oriented point in the path do
    for each vertex in the 2D shape do
        Add the vertex position, based on the oriented point
        Add the normal direction, based on the oriented point
    end for
end for
for each segment do
    for each line in the 2D shape do
        Add tow triangles with vertex indices based on the line indices
    end for
end for
```

Then the next iteration is started. The number of segments is the number of oriented points in the extrusion path minus one. For creating each segment, we iterate over the lines which are stored in the 2D shape lines list. Each line is defined by two integers (later called line indices) which are the vertex indices of the start and the end vertices. Triangles are generally defined by only referencing the concerned vertices in the vertices list which should be included in the triangle. So for each segment, two triangles are added with vertex indices based on the line indices.

#### Created Extrusion Objects

The `GenerateMeshAlongPath` component describes the behaviour of each extrusion object. It contains a reference to an `ExtrusionModel` instance which stores the associated shape and path data.

It also handles the `LineRenderer` components related to the shape and path game objects. There are four `LineRenderers`. Both, the shape as well as the path, have two representatives for 2D space and 3D space. The 2D `LineRenderer` is positioned on the drawing canvas whereas the 3D `LineRenderer` is placed directly on the extrusion object in 3D space. The 3D `LineRenderers` are essential for creating a connection between the 2D drawing and the 3D object. Moreover, the `GenerateMeshAlongPath` component contains the `extrudeMesh` method, which generates a mesh out of a custom 2D shape and an extrusion path. This method is called every time after a vertex has been added to the path.

### 4.3.4 Pen Interaction

The *Microsoft Surface Pen* is used for interaction with the user interface. As mentioned in section 3.4.2, the pen has buttons that could also be used for certain features such as translation in z-direction or deleting objects with the eraser button. Unfortunately, the button events are not supported by Unity.

Hence, only the touch position of the pen is used and further processed in the tablet application. The raw input position of the pen is provided in screen coordinates and is converted into world space coordinates before it is further processed.

### 4.3.5 Planar Visualization of Extruded Objects

As shown in figure 4.12, the user interface of the tablet application includes a preview window component to provide the user with a planar presentation of the 3D scene. The figure 4.5 presents two images of the preview window in the prototype. Users can view the extruded objects in this window simultaneously while creating them. The preview window is also used for object selection by clicking on an object. The preview window has been realized with an additional camera in the scene. If an input event occurs, the *GameController* component checks, if the raw input position, which is given in screen space and therefore pixel coordinates, is contained in the *pixelRect* of the preview camera. The *pixelRect* of a camera describes where on the screen the camera is rendered in pixel coordinates. If the input position is contained in the *pixelRect*, a ray will be cast into the scene, and if this ray hits an object, it is selected.

Furthermore, the preview window only reveals game objects which are placed in the *Scene3D* or *Scene2D3D* layer. Game objects, which are placed in the *Scene2D3D* layer, are visible to both cameras (main and preview camera). The main camera displays the drawing canvas and game objects which should only be visible in the 2D view, for example, the shape line and extrusion path line. This is why the game objects, presented by the preview camera, are always placed in the *Scene2D* layer.

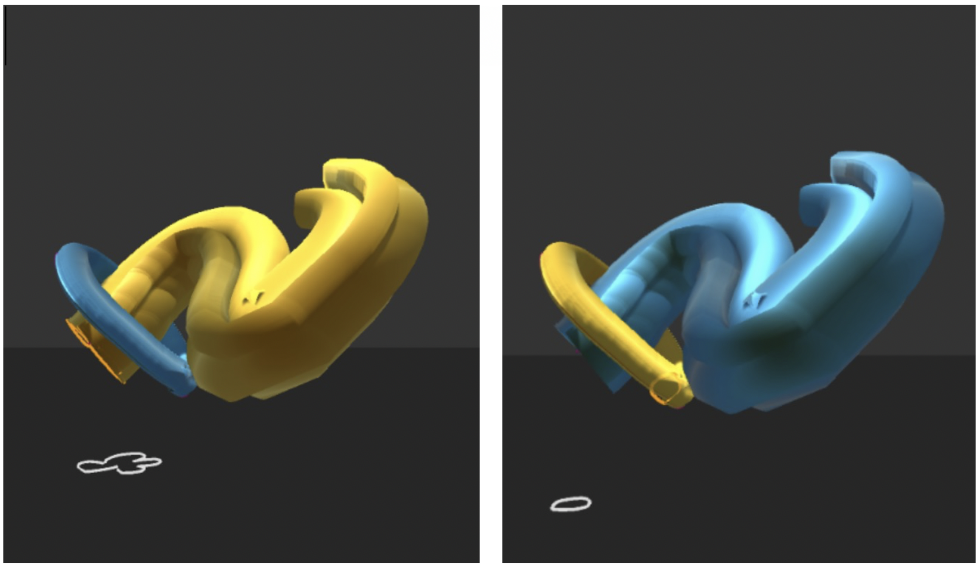


**Figure 4.5:** Planar visualization of extruded objects and object selection in the preview window.

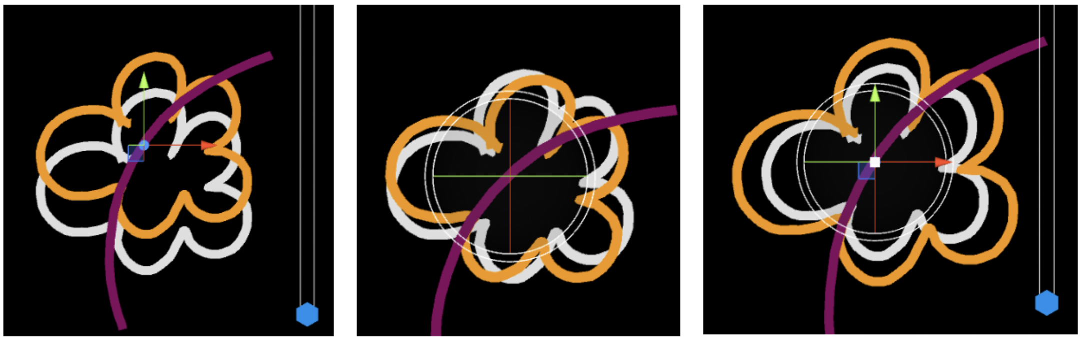


**Figure 4.6:** The translation, rotation, scaling and universal gizmo are placed at the center of the custom 2D shape.

### 4.3.6 Basic Object Transformations

The basic object transformations, which are translation, rotation and scaling, have been realized with the help of an asset[6], available on the *Unity Asset Store*. The *Runtime Transform Gizmo* (RTG) asset allows incorporating transform gizmos into the application during runtime. In figure 4.6, all gizmos are shown. The transform gizmos provide translation, rotation and uniform scaling along all three axes which have the color red for the x-axis, green for the y-axis and blue for the z-axis. As shown in the third image in figure 4.6, the RTG asset provides a universal gizmo that combines translation,

[6] Runtime Tranform Gizmo: https://assetstore.unity.com/packages/tools/modeling/runtime-transform-gizmos-125537, last visited on 11.6.2019

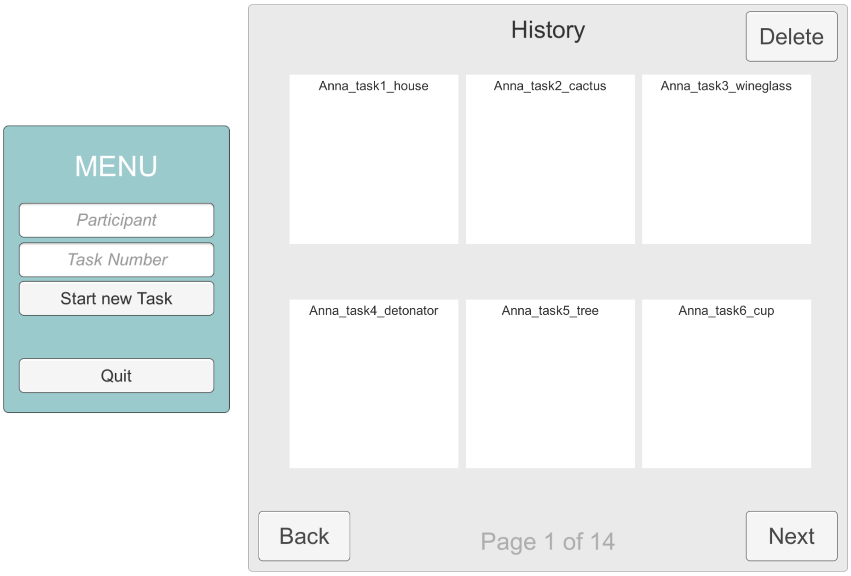


**Figure 4.7:** The file management menu for saving and loading scenes and for documenting during the user study.

rotation, and uniform scaling (white rectangle in the center) in one gizmo.

A disadvantage of the translation and scaling gizmo is that the blue axis cannot be used because the canvas is fixed in the xy-plane. Therefore, an additional slider at the right border has been added to enable translation along the z-axis.

Furthermore, object transformations are executed relative to the pivot point. For example, if the pivot point is set to the origin and the object rotates, the object will be rotated around the origin. Usually, the pivot point matches the center of the whole object. In the prototype, however, a design decision was to set the pivot point always to the center of the drawn 2D shape, because the user interface is two-dimensional and the three-dimensional objects are only visible in the preview window and not in the drawing canvas. Therefore, it looks more natural to set the pivot point to the center of the visible shape and not to the invisible object.

### 4.3.7 Save and Load File Management

The save and load functionality, which is needed for documenting the created scenes during the user study, has been realized with the help of the *SaveIsEasy* asset[7] from the *Unity Asset Store*. The figure 4.7 presents the file management interface. The user can enter the name of the participant and the current task number. By clicking on the *Start New Task* button, the user can start a new task session. By selecting one of the saved scenes in the *History* panel, the appropriate scene will be loaded and available to view again. By pressing the *Quit* button, the application will be closed.

[7] SaveIsEasy: https://assetstore.unity.com/packages/tools/input-management/save-is-easy-57432, last visited on 11.6.2019

## 4.4 System Architecture of the HoloLens Application

The architecture of the HoloLens application is shown in figure 4.8. The figure includes the major classes of the HoloLens application and how they are related to each other. In the following section, system components such as the network connection and the stereoscopic visualization of extruded objects will also be described in detail.

The structure of the HoloLens application is similar to the Surface application. They both have a networking component, in which either the server (Surface) or the client (HoloLens) will be started. Also, they both have an extrusion component, a basic object transformation component and a UI controller component that manages the created extrusion objects as well as the UI events.

The main difference is, that the Surface application receives user input directly through the touch surface, whereas the HoloLens application receives the user input events over the network. Furthermore, the HoloLens application does not provide a planar visualization of extruded objects or a file management system.

### 4.4.1 UI Controller

The messages, which are sent from the server component on the tablet application, are received by the `WebSocketClient` component in the HoloLens application. After deserializing the JSON object back into a `MsgBundle` object, the message will be queued as an event in the `DrawingController` component, which is the UI controller.

The `DrawingController` executes these events and calls methods appropriate to the editor mode code inside the `MsgBundle`. For example, if the code is four, the `DrawingController` will call the method `addNewExtrusionObj`.

Hence, the `DrawingController` dispatches the user input received from the server and calls the related methods.

Similar to the counterpart `GameController` on the tablet, the `DrawingController` is also responsible for holding the created extrusion objects and keeping track of the currently selected object.

As shown in figure 4.8, the `GenerateMeshAlongPath` class and its related components are equal to the components in the surface application.

### 4.4.2 Stereoscopic Visualization of Extruded Objects

The drawing canvas, in which the user draws his shapes and extrusion paths, is defined in the xy-plane. The origin of this coordinate system is the bottom left corner in pixel coordinates, and the top right corner is *(Screen.width, Screen.height)*.

The origin of the *Image Target* is at the center of it. The *Image Target* is placed at the upper left corner, as shown in figure 4.12. The width of the *Image Target* is one and the height dependent on the format of the marker. Furthermore, the coordinate system of the drawing canvas is given in world space coordinates. Hence, a conversion needs to be calculated in order to get the correct offset for the stereoscopic visualization. The stereoscopic visualization should be placed precisely on top of the drawn 2D shape, as shown in figure 4.9.

The goal is to find out the offset between the origin point of the marker (center) and the origin point of the drawing canvas (bottom left corner). The offset needs to be

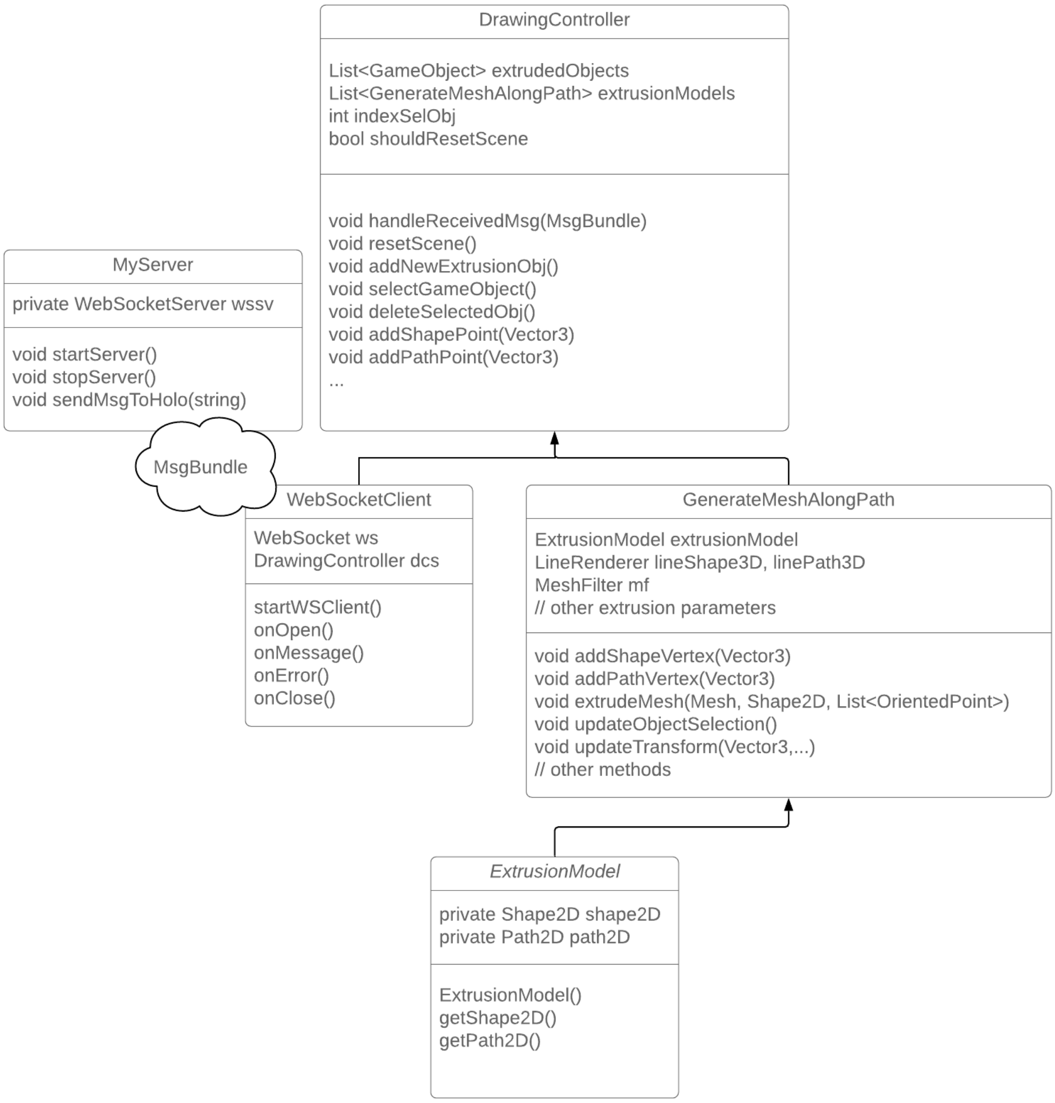

DrawingController
List<GameObject> extrudedObjects
List<GenerateMeshAlongPath> extrusionModels
int indexSelObj
bool shouldResetScene
void handleReceivedMsg(MsgBundle)
void resetScene()
void addNewExtrusionObj()
void selectGameObject()
void deleteSelectedObj()
void addShapePoint(Vector3)
void addPathPoint(Vector3)
...
MyServer
private WebSocketServer wssv
void startServer()
void stopServer()
void sendMsgToHolo(string)
MsgBundle
WebSocketClient
WebSocket ws
DrawingController dcs
startWSClient()
onOpen()
onMessage()
onError()
onClose()
GenerateMeshAlongPath
ExtrusionModel extrusionModel
LineRenderer lineShape3D, linePath3D
MeshFilter mf
// other extrusion parameters
void addShapeVertex(Vector3)
void addPathVertex(Vector3)
void extrudeMesh(Mesh, Shape2D, List<OrientedPoint>)
void updateObjectSelection()
void updateTransform(Vector3,...)
// other methods
ExtrusionModel
private Shape2D shape2D
private Path2D path2D
ExtrusionModel()
getShape2D()
getPath2D()

**Figure 4.8:** Class diagram representing the system architecture of the HoloLens application and the relations of the main components.

calculated in the coordinate system of the marker because the stereoscopic visualization is placed with this coordinate system.

For the conversion, the width and the height of the *Image Target* itself and the preview window have been measured in centimeters. The measured height of the marker is, in this case, 5.4 centimeters and the height of the preview window is 9.8 centimeters. Because the origin of the marker is at the center of the marker, half of 5.4 (2.7 centimeters) are added to the height of the preview window.

Now, we can set the marker heights (0.7 units in Unity and 5.4 centimeters in reality) in a relation which equals the relation of the searched variable (height of the preview

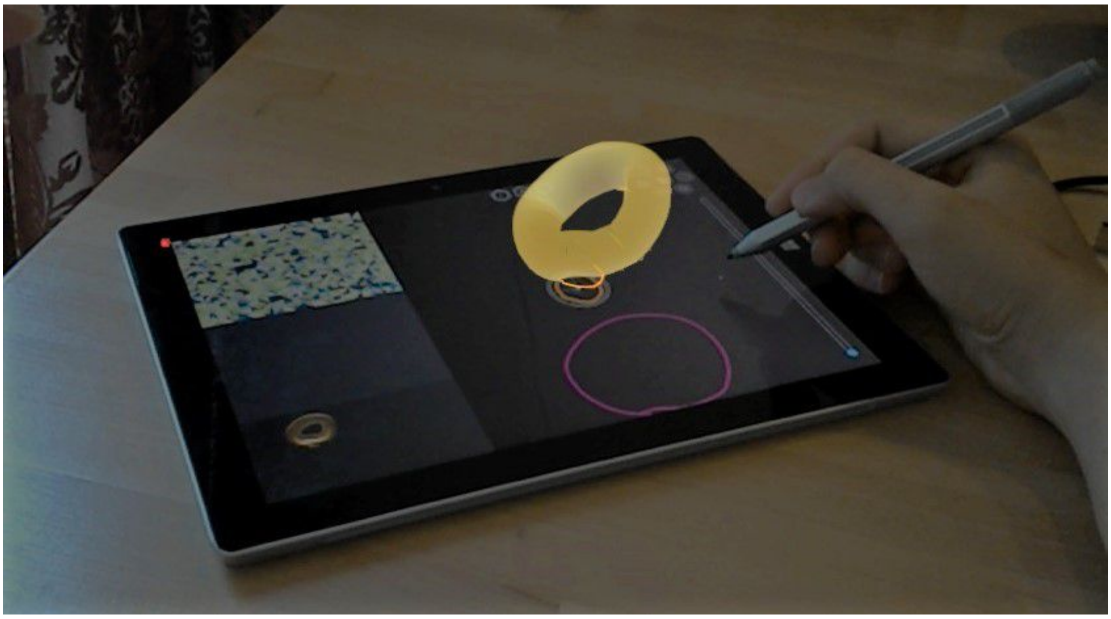


**Figure 4.9:** A stereoscopic visualization of a virtual extrusion object created by the user with two drawn lines (shape and extrusion path).

window plus the half of the marker in units) and the same height in centimeters. After solving the equation, the searched heights appeared to be approximately 1.62 units in Unity. Hence, the stereoscopic visualization is translated along the y-axis. The offset along the x-axis is minus the half of the marker width, because both, the marker and the preview window are placed on the left side.

## 4.5 Prototype Workflow

In figure 4.11, a sequence diagram is presented in order to give a better overview of the prototype's workflow.

The sequence diagram shows the user, who is interacting with the surface application. The major components of the surface application are presented on the left side of the diagram and contain the UI elements such as buttons, gizmo handles and sliders but also the `GameController` component, the `WebSocket` Server, the `GizmoManager` and the `FileManager`.

The diagram assumes that the connection between the server and the client has already been established. The user starts with pressing on the add button to create a new object. The server is informed about this event and sends the appropriate message to the `WebSocketClient` component, which is part of the HoloLens application so that the HoloLens application adds an object as well.

Then the user continues with drawing a line. Since the `GameController` manages pen input events, he is notified and also propagates a message to the client. Further progress is similar to both applications. The selected object is chosen from the list of objects, which is stored in the `GameController` as well as in the `DrawingController`.

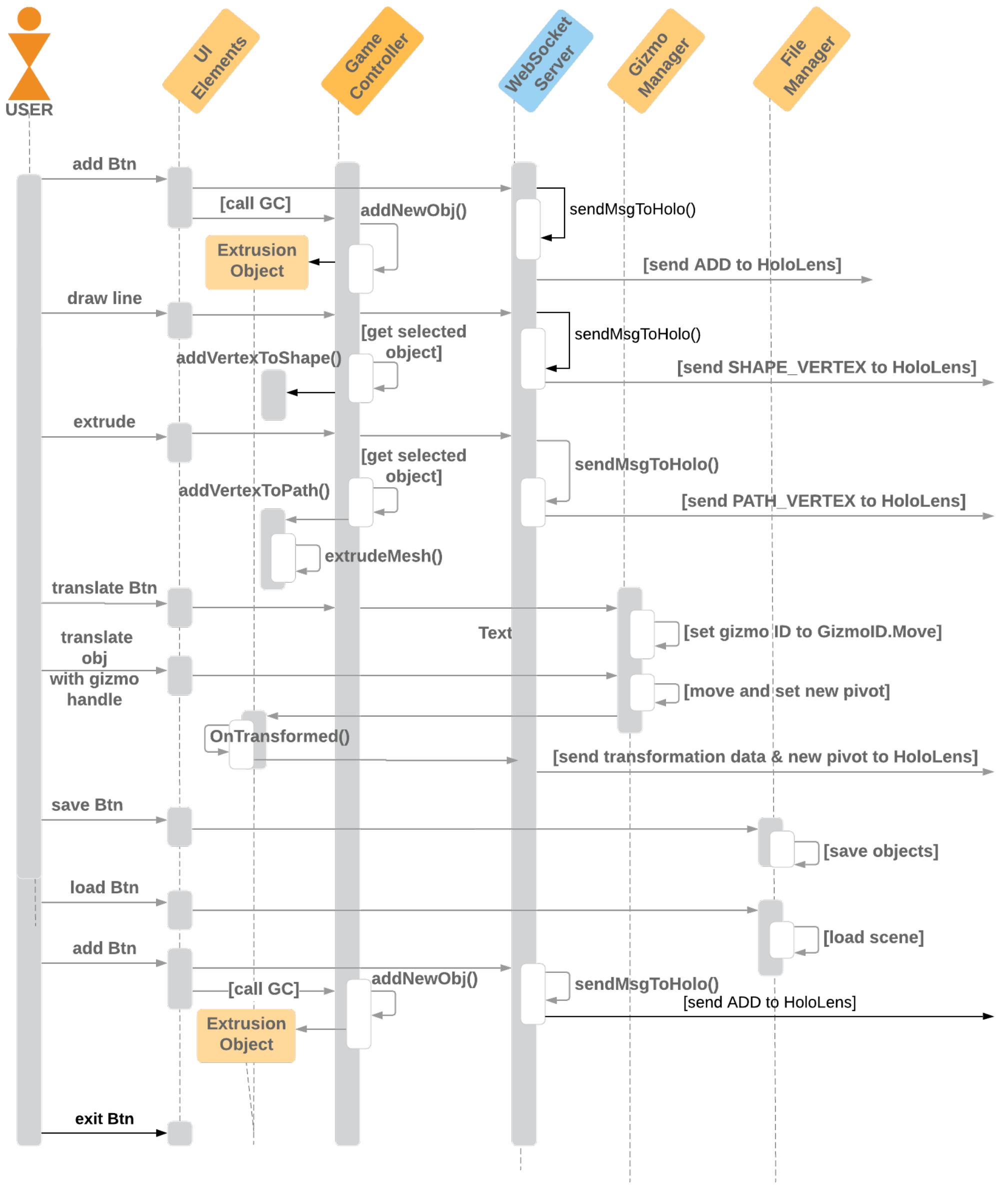


**Figure 4.10:** Surface part in the sequence diagram

The method `addVertexToShape` is called in the related extrusion object.

The sequence diagram also illustrates the workflow in case of extrusion, translating an object and saving and loading a scene.

## 4.6 User Interface

The user interface of the Surface application is shown in figure 4.12, and it consists of multiple components.

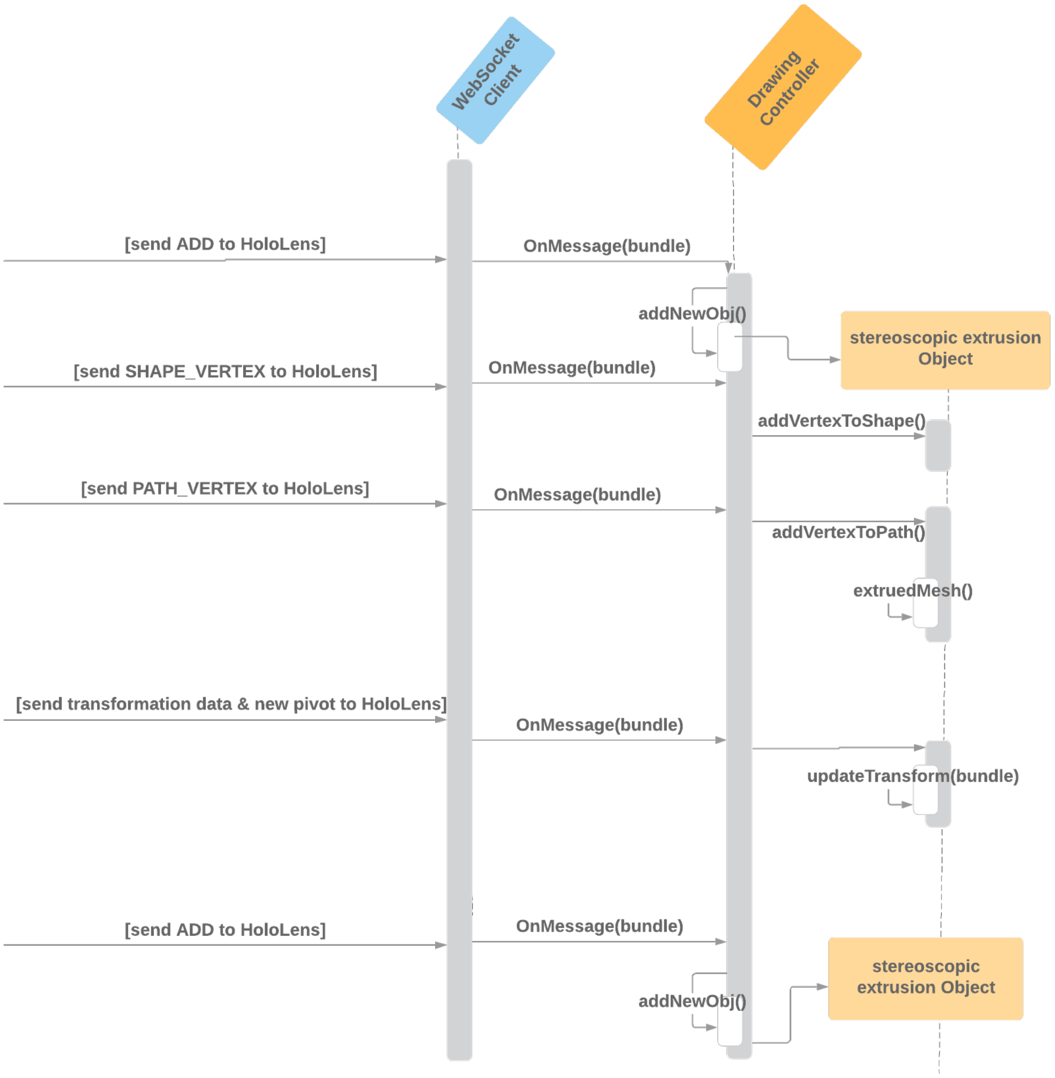


**Figure 4.11:** HoloLens part in the sequence diagram

First of all, there is an AR marker in the upper left corner, which is needed for tracking the tablet and for creating a coordinate system. The coordinate system can be referenced when placing virtual objects on the marker, which is placed on the tablet. The marker has been realized with an *Image Target* from the *Vuforia* SDK and has the highest rank when it comes to optimizing target detection and tracking stability. The motive for the *Image Target* was downloaded from an online blog by Dario Mazzanti [34][8].

Underneath the marker, there is the preview window. The preview window presents a planar visualization of the extruded objects to the user. It has been realized with a

[8] Source of the image: https://i0.wp.com/www.dariomazzanti.com/wp-content/uploads/2017/10/generated_1.png?w=1280&ssl=1

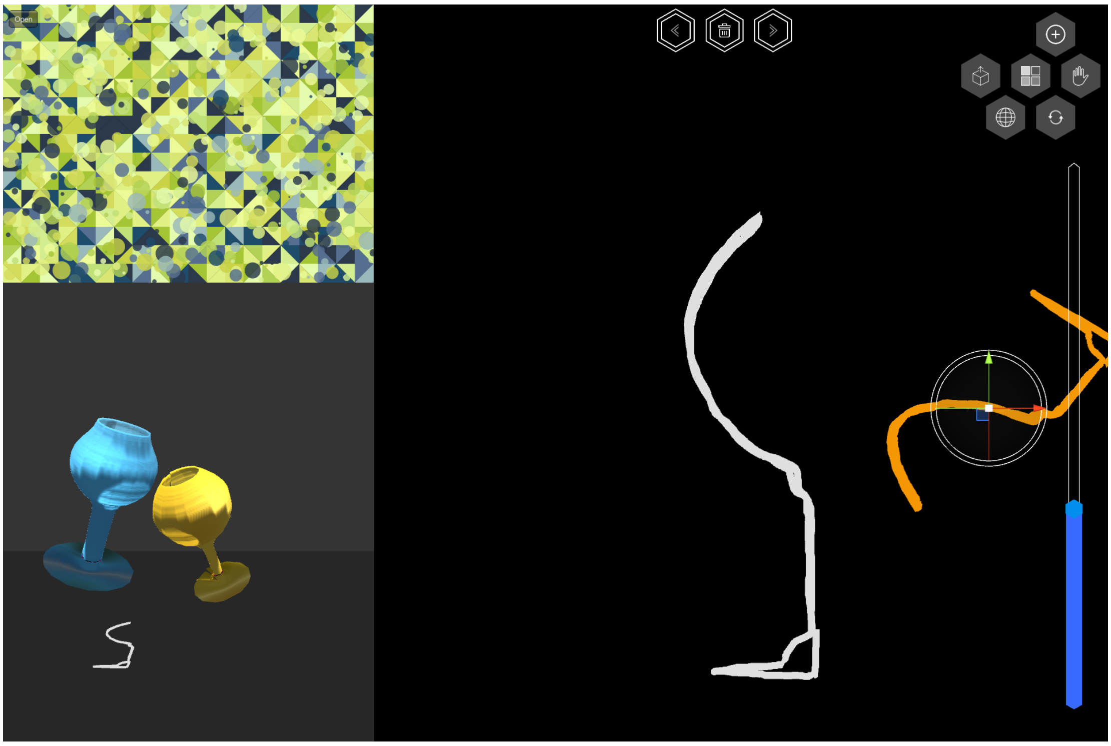


**Figure 4.12:** User interface of the *Microsoft Surface* application

second camera in Unity, which is looking at the scene of created objects. It is showing game objects, which belong to the 3D scene view and it also includes the 2D shapes which have been drawn on the canvas. Nevertheless, the main camera, which is looking at the drawing canvas, does not show the extrusion and the resulting objects, but only the drawn lines and forms (2D as well as 3D). The user can also select objects in the preview window to be able to manipulate them in further progress.

The largest area in the user interface is reserved for the drawing canvas because the user needs space for drawing. After placing the virtual extrusion objects relative to the marker and on the tablet, the stereoscopic visualization appeared to be better visible if the background is dark instead of light which was the reason for the black background of the drawing canvas.

Furthermore, multiple buttons are placed in the upper right corner and on the top of the screen for basic object transformations such as translation (hand icon) and rotation (circular arrow icon) as well as basic features such as extrude (cube with arrow up), add (plus), delete (dustbin), undo (arrow left) and redo (arrow right). There is also an extra button for universal transformation (circular grid) which offers a gizmo in which all three basic object transformations are combined. Therefore, it also includes uniform scaling functionality.

When the Surface application starts, the user has to press the *Add* button in order to add a new object to the scene. Afterwards, he can start drawing on the drawing canvas. If he wants to extrude this shape, he has to press the *Extrude* button and draw the extrusion path on the drawing canvas. To be able to distinguish between the two lines, the color of the shape has been chosen to be orange, and the extrusion path is purple.

If the user wants to translate the object, he has to press the *Translate* button. In this case, the gizmo will appear at the center point of the shape, which is set as the pivot point of the object. Because translation along the z-axis is not possible on a two-dimensional surface, a slider has been added on the right side of the screen. The slider has the color blue to indicate the user, that it is related to the z-axis. When translating the object in any way, the original position and size are marked with a white line.

If the user has made a mistake, he can undo the last 50 steps as well as reapply them again by pressing the *Redo* button.

In the upper left corner, a menu is hidden. When pressing the *Open* button, a panel opens, and the user can press *Save* in order to save the scene and press *Export .obj* to export the created object and store it outside of the application, as shown in the screenshots section in Appendix A. When pressing on the *Menu* button, the user enters the file management system and can select and load a scene or start a new task.

## 4.7 Build and Deployment

The following section describes how the Surface application, as well as the HoloLens application, are built and deployed to the corresponding devices.

### 4.7.1 *Microsoft Surface*

The *Microsoft Surface* application is built as a *Windows Standalone* application. The platform can be chosen in the *Build Settings* in Unity. The target platform in the Build Settings is set to *Windows*, and the architecture property is set to *x86_64*. After building the project, the folder with the *.exe* file can be moved to the *Microsoft Surface* device and can be executed.

### 4.7.2 *Microsoft HoloLens*

The application for the *Microsoft HoloLens* is built as a *Universal Windows Platform* application. Similar to the *Microsoft Surface* application, this platform can be selected in the *Build Settings*. The target device option is set to HoloLens and the build type to *D3D*. The SDK and *Visual Studio* versions are set to the latest installed.

After building the project, a *Visual Studio Solution* has been generated, which is then opened. The solution configuration is set to *Release*, the solution platform to *x86* and the *Remote Machine* is selected for execution. For deployment, the HoloLens needs to be in the same network.

# Chapter 5

# Evaluation

In the previous chapter *Technical Implementation*, the prototype has been described in detail. The following chapter evaluates this prototype with the help of a user study. The modality of how the user study has been executed will be described. Furthermore, the results, the interpretation of the results, and a final discussion and an answer to the research question will also be included in this chapter.

## 5.1 Research Question

The prototype related to this master thesis consists of a tablet-based user interface on which the user can draw 2D shapes. These shapes can be extruded along a custom extrusion path to generate a 3D object out of 2D forms. The generated 3D object is displayed either planar on the tablet or is presented as a stereoscopic visualization, which is attached and directly placed on the tablet. The stereoscopic augmentation is shown by a see-through HMD the user is wearing while creating objects. The implemented prototype is used to investigate the following research question.

> How does AR-based stereoscopic visualization support the user in creating extrusion objects, and what are the advantages and disadvantages of it?

## 5.2 Hypotheses

With the help of a user study, the resulting prototype will be evaluated in terms of advantages and disadvantages when using AR for stereoscopic visualization. Therefore, the user study aims to compare AR-based stereoscopic visualization directly placed on the tablet with a conventional planar presentation of 3D objects.

For the comparison of these two variants, four hypotheses have been defined to set up the objectives for the user study. The main objectives of the user study are to verify the following hypotheses.

**H1.** The participants are faster in creating and reconstructing a given 3D object when a stereoscopic visualization is available.

**H2.** The participants are making fewer mistakes when a stereoscopic visualization is available.

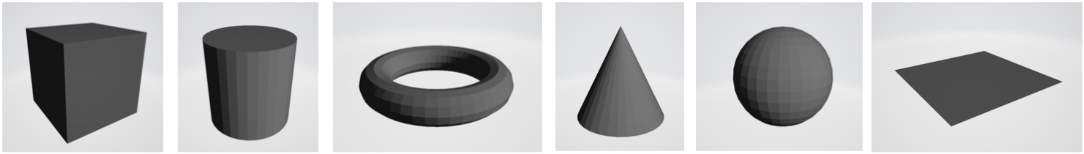


**Figure 5.1:** Preliminary training tasks for reducing the learning effect on user study tasks.

**H3.** The participants are creating and reconstructing a given 3D object more precisely when a stereoscopic visualization is available.

**H4.** It is easier for the participants to work with 3D objects in 3D space with the help of a stereoscopic visualization compared to a planar 3D visualization.

## 5.3 User Study

To conduct the user study, it is necessary to prepare particular parts of the study in advance for example the tasks each participant has to perform, a common test procedure which is similar to all participants, the test environment, the structure of the log files and the data and the evaluation modalities to define how the gathered data will be processed afterwards.

### 5.3.1 Tasks

It is a logical implication that the participants will become faster the longer they are interacting with the user interface because they get used to the interface. This circumstance is well-known as the learning effect [22]. The user study session starts with a training session to reduce this learning effect. The participants are asked to rebuild simple objects which are shown in figure 5.1.

Six tasks have been prepared to compare planar visualization of 3D objects on the tablet with stereoscopic visualization placed on the tablet presented by the HoloLens, which are shown in figure 5.2.

Each task was presented to the participant printed on paper and with half the size of an A4 format. Each task consists of a single 3D object, and the participants were asked to recreate these 3D objects. The participants generate multiple sub-objects and transform (translate, rotate, scale) the sub-objects to form the result similar to the sample object in the figure 5.2.

### 5.3.2 Participants

For the user study, twelve participants have been recruited to perform the tasks, which have been introduced in section 5.3.1.

Two of them are female, and ten of the participants are male, and their age ranges from 22 to 35. Regarding the background of the subjects, three participants have studied in the field of hardware and software engineering, four participants were recruited from

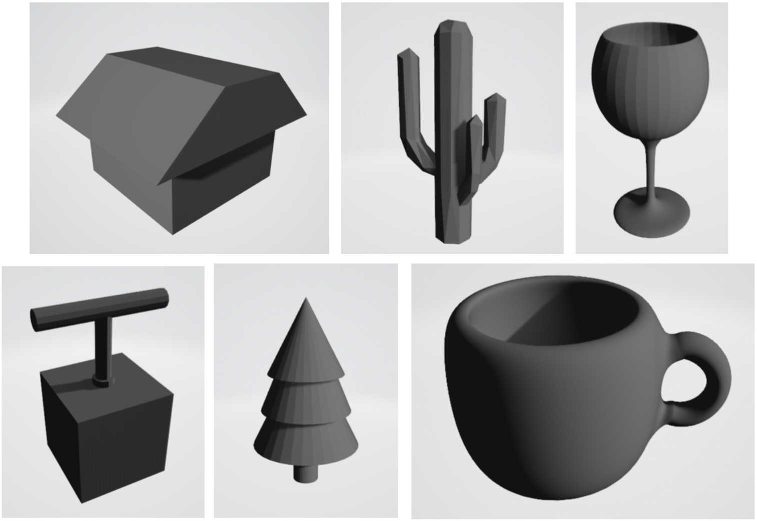


**Figure 5.2:** All of the six objects were recreated by each user. The first test group fulfilled the first three objects (upper row) without the HoloLens and the last three (lower row) with the HoloLens. The second test group switched the order of visualization modality and started with using the HoloLens.

the mobile and web development sector, three participants are operating in the management sector, and two participants are working as machinists in the metal construction sector.

All participants, except for one, do not have any experience with AR technologies such as the Microsoft HoloLens. They are all classified as amateurs. The one with experience in the field of AR is classified with advanced. He has already developed an application for the Microsoft HoloLens and therefore could study the device in more detail. He also has experience in wearing and using the HoloLens, whereas all the others have not.

Half of the participants had normal vision, and the vision of the other half was corrected to normal. Wearing glasses was not a problem while wearing and operating with the HoloLens. Two participants were left-handed.

### 5.3.3 Test Procedure

The test procedure declares how a test session should be fulfilled. A test session consists of an introduction part, a training unit (to reduce the learning effect on the subsequent task session), six task sessions, a qualitative evaluation unit and a short interview with the participant.

Every test session starts with documenting personal data of the tested subject (gender, age, profession, educational background, degree of experience related to Mixed Reality technologies, vision disabilities, and handedness). Afterwards, the participant was introduced to the topic and the aim of the user study.

Next, the training unit begins, and the supervisor explains the tablet-based user interface and all its sections and features to the participant. Furthermore, the supervisor explains extrusion as the input modality to the participant and how to use it for creating objects. Afterwards, the supervisor assigns the training tasks to the subject which are shown in figure 5.1. The training tasks consist of basic object primitives to clarify extrusion based on simple objects. If the participant feels confident enough to handle the user interface and object creation and transformation, the test session continues with the six task sessions.

The subject will be given six different tasks, for instance, a house. The subject has to rebuild the objects in different scenarios. Once with an AR-based stereoscopic visualization on the tablet-based surface and once without stereoscopic but with planar visualization on the tablet-based surface. To be able to compare planar with stereoscopic visualization, the participants have to perform three tasks with and three tasks without the HoloLens.

It is reasonable that the subjects are getting better with every task they fulfill. Consequently, the gained experience in the following tasks will influence, for example, the operation times related to the current visualization scenario. The participants have been split into two test groups of six participants each. The order of the two visualization scenarios has been counterbalanced to reduce the learning effect. The first test group (#1, #2, #3, #4, #5, #6) performs the first three tasks without the HoloLens and the last three tasks with the HoloLens. The second test group (#7, #8, #9, #10, #11, #12) starts with the same three tasks but is now using the HoloLens. In this way, the impact of the learning effect on the comparison of stereoscopic and planar visualization can be reduced.

Because it is subjective to decide if a task is fulfilled or not, every participant is allowed to stop the task session whenever they think that their recreation of the sample object is finished.

After the participant stops the task session, the supervisor saves the scene, exports the resulting object for documentation purposes, and starts the next task session.

When the sixth task has been finished, the user has to fill out the NASA-TLX [15][1] questionnaire, which will be needed for the qualitative evaluation. Finally, there will be a short interview in which the supervisor asks a few questions, receives feedback, and discusses the system with the participant.

### 5.3.4 Setup

The setup for the user study is shown in figure 5.3. The participant was presented the task he had to complete on a paper which is also visible in figure 5.3. The sample object in this task session was the cup. Therefore the participant had to recreate the cup. The subject was recreating the sample object by drawing shapes and extrusion paths on the Microsoft Surface with the Microsoft Surface pen. He had to complete this task in the HoloLens scenario. Therefore, he was wearing the *Microsoft HoloLens 1* which presented the stereoscopic visualization tho him which was placed on the tablet.

All test sessions were conducted indoors, in the same room, and with similar lighting conditions. There was no direct sunlight but constant light in the room to avoid

[1] For the qualitative evaluation of the user study, the adapted version of the NASA-TLX questionnaire in German was used which was published by the Interaction Design Group of the Institute for Industrial Design in Magdeburg-Stendal [33].

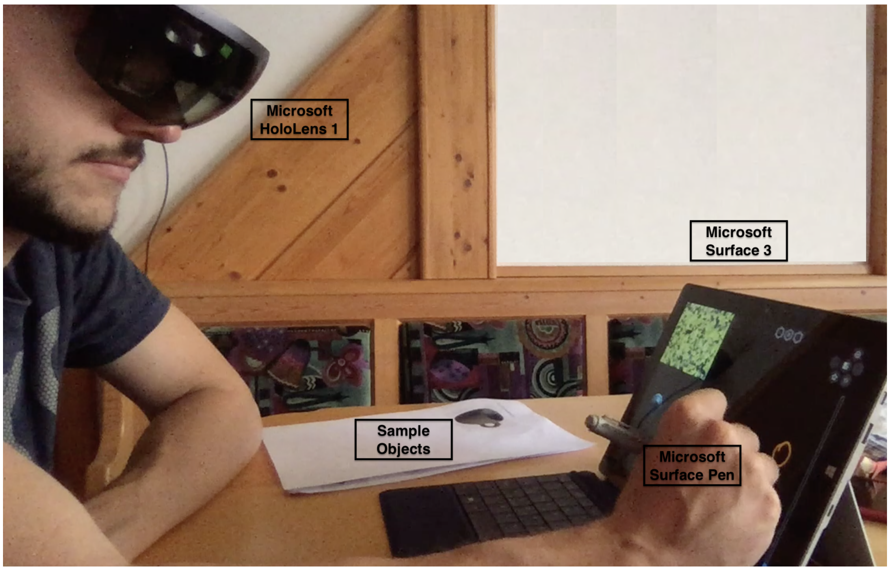


**Figure 5.3:** The setup for the user study includes the *Microsoft Surface 3* for providing the user interface, the corresponding *Microsoft Surface Pen* the participant is drawing with and the *Microsoft HoloLens* which presents the stereoscopic visualization to the participant.

**Table 5.1:** Excerpt of generated log file per participant and per task.

| Timestamp | Input Coord. | UI Area | Category | Action |
|---|---|---|---|---|
| 3ms | [12.2, 15.0] | SP | D | Selection |
| 10ms | [1.0, 2.2] | D | C | Extrusion |

unnecessary reflections, which would disturb the participant and also hinder the target recognition for the HoloLens because of the reflections on the *Microsoft Surface*. The room was quiet, clean, and tidy to avoid deflecting the participant´s attention away from the task.

### 5.3.5 Data Logging and Documentation

To address the objectives of the user study, which have been declared in section 5.2, data is logged during each user study session. The logging data is created for each participant and each task. Each input the participant enters is logged into a log file. The structure of the log file is shown in table 5.1. The timestamp logs the point in time in milliseconds (since application start) when an input is detected. The input coordinates are representing the coordinates on the screen where the input has been detected.

The UI area declares in which part of the user interface the input has occurred (drawing section (D), scene preview section (SP), tool buttons (T), image target (IT), save and load section (SL)). The values of the modeling category are defined in table 5.2 which were presented in the papers [24] and [23]. The action attribute is an accurate

**Table 5.2:** Modeling operation categories which were presented in the papers [24] and [23].

| Operation Category | Code | Operation |
|---|---|---|
| Navigation | N | Navigation for better view of the model, assessment of a scene, or walk-through. |
| 2D Drawing | D | 2D drawing activities, e.g. drawing, selecting, or editing 2D. |
| 3D Creation | C | Add 3D primitives, extrude 2D contours, clone existing ones. |
| Modification | O | Sculpt or extrude to change shape. |
| Manipulation | A | Resize, extrude to resize, rotate, move, or remove. |
| Material | T | Apply or change texture or color. |
| Miscellaneous | I | Any activities that are not directly related to changing the geometry. |

description of the modeling category related to the input node. The log file is used for quantitative measurements regarding task completion times, operation times, and the error rate.

Furthermore, the resulting user-generated objects are exported, and the corresponding scenes are saved for documentation to be able to review and rate them afterwards. The test sessions have also been video recorded to facilitate the evaluation of the log files. The recordings are also important for estimating if the participants, for example, paused interaction because of an interruption or because he was viewing his generated object.

## 5.4 Results and Observations

The following section examines the results of the user study in terms of time tracking, error rate, precision, and qualitative measurements.

### 5.4.1 Time Tracking

The time is tracked within the log file in milliseconds and is used for calculating the task completion times for each user. Furthermore, the timestamps are also used for calculating the temporal ratio between tablet and HoloLens for investigating which visualization scenario is faster.

### Task Completion Times

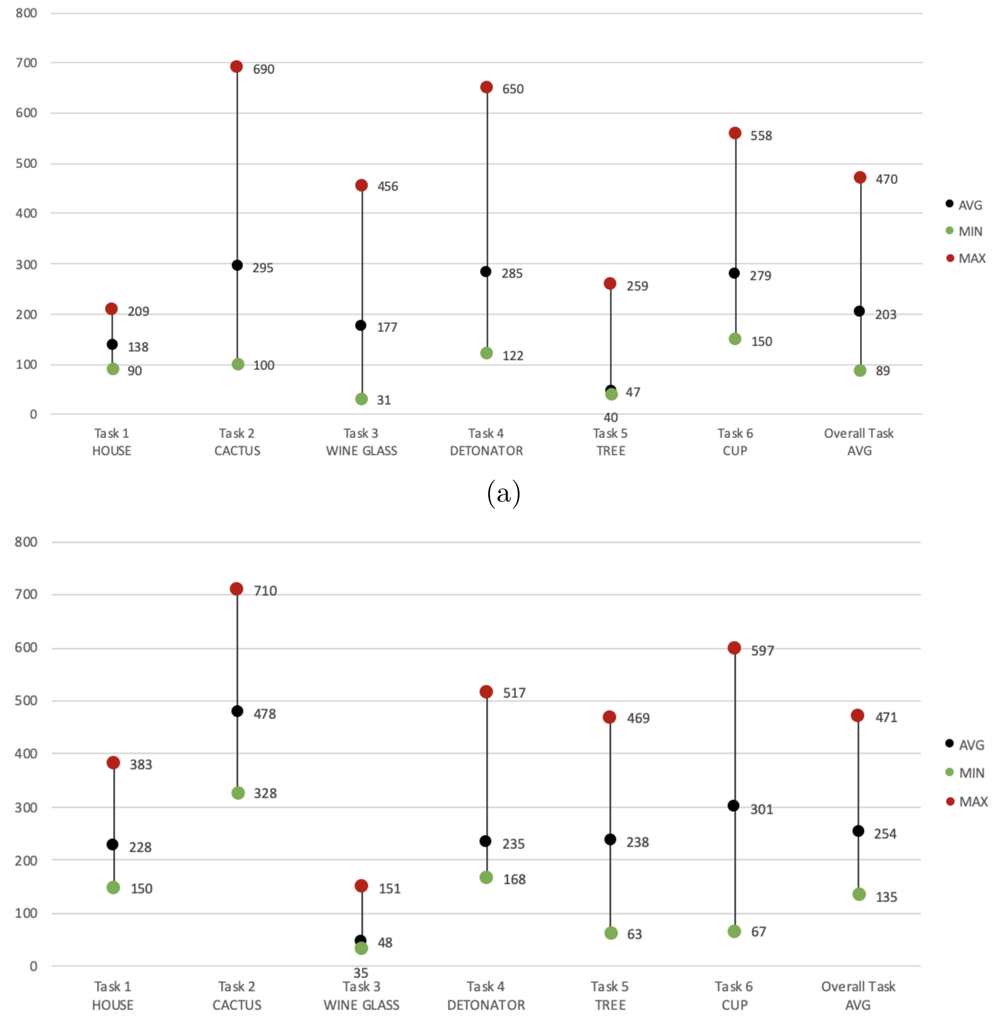
In general, most of the subjects were able to perform all six tasks in a reasonable amount of time which was 3.3 minutes (203 seconds) on average with the tablet and 4.2 minutes (254 seconds) on average with the HoloLens. Figure 5.4 illustrates the maximum, minimum, and average task completion times per task in seconds for the tablet (a) and the HoloLens (b) scenario. Participants in the tablet scenario completed the tasks with a maximum of 11.5 minutes (690 seconds) and with a minimum of 31 seconds whereas participants in the HoloLens scenario completed the tasks with a maximum of 11.8 minutes (710 seconds) and with a minimum of 35 seconds. It is visible that three participants stand out because they completed tasks with a task completion time above ten minutes (600 seconds). During the user study it appeared, that reasons for a long task completion time are the overlay of the stereoscopic visualization directly on the user interface on the one hand and on the other hand, problems with understanding the functionality of extrusion for creating 3D objects.

A direct comparison between the planar (tablet) and the stereoscopic (HoloLens) visualization scenarios has been illustrated in figure 5.5 and 5.6. Figure 5.5 presents the task completion times concerning tablet and HoloLens. Hence, all the task completion times of the participants, which have performed the tasks with the tablet, have been summed up and then the relative percentage of the total time per task (sum of task completion times of all subjects in one task) has been calculated and vice versa for the HoloLens.

The relations in figure 5.5 indicate a trend that the participants are taking more time in performing the tasks when wearing the HoloLens. At an average, participants wearing the HoloLens are 5.8% slower than the participants without wearing it. In figure 5.6, the relations of tablet and HoloLens per subject are presented. In a short interview after the task sessions, a participant, who was faster when wearing the HoloLens, stated, that he adjudged working only with the tablet to be better for him. He said he was focusing more on the planar visualization than on the stereoscopic visualization because the stereoscopic visualization is placed directly on the tablet and therefore occludes his workspace, which is the user interface on the tablet. However, it is also necessary to mention that this participant, in particular, is already very experienced when it comes to working with object modeling tools which means that he is used to planar visualizations of 3D objects.

Another participant who was standing out during the user study was subject #12. This participant has stated that he has no experience with object modeling tools, AR technologies, and extrusion. He also stated in the interview that he enjoyed working with the HoloLens more because he had a better view of the object, and it was much easier to view the object from different perspectives. Nevertheless, it took the participant longer to perform the task because the participant was spending more time in observing the stereoscopic visualization.

Overall, nine out of twelve participants stated in the interview that they enjoyed performing the tasks more when wearing the HoloLens. There is a slight trend that participants are spending more time viewing the object when there is a stereoscopic visualization available. This trend is also visible in figure 5.7.



(a)

(b)

**Figure 5.4:** Minimum, average and maximum task completion times per task in seconds of Tablet (a) and HoloLens (b).

Consequently, the hypothesis H1 from section 5.2, which states that participants are faster in creating and reconstructing a given 3D object when wearing the HoloLens, is therefore disproved.

### Mean Operation Times

A task is performed using different operations, which are 2D drawing, extrusion, translation, rotation, scaling, selection, and pause. The category pause is composed of user interface interaction because pressing a UI button is only a point in time and the time until the subject performs a continuous operation which lasts over a period of time (object transformations, drawing, extrusion) can be considered as a paused state because the participant is not interacting with the UI in this period.

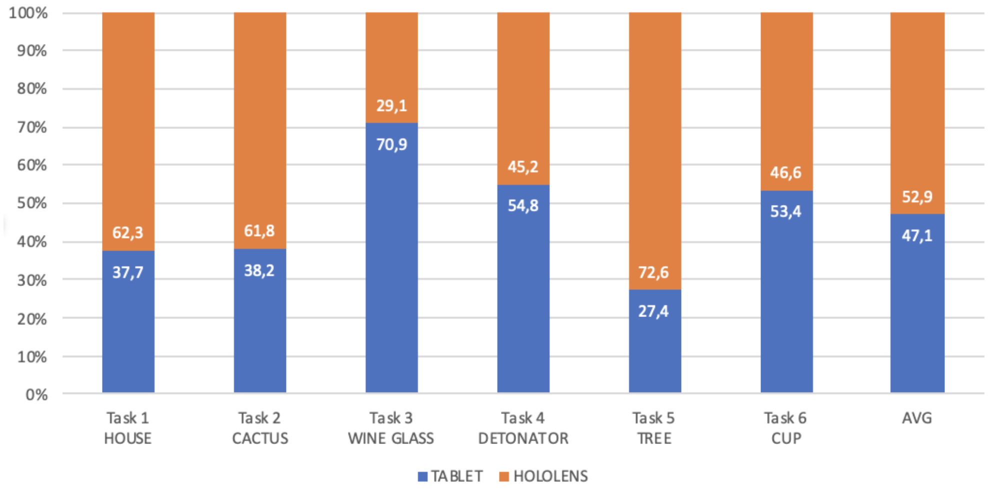


**Figure 5.5:** Tablet and HoloLens ratio regarding the task completion times per task.

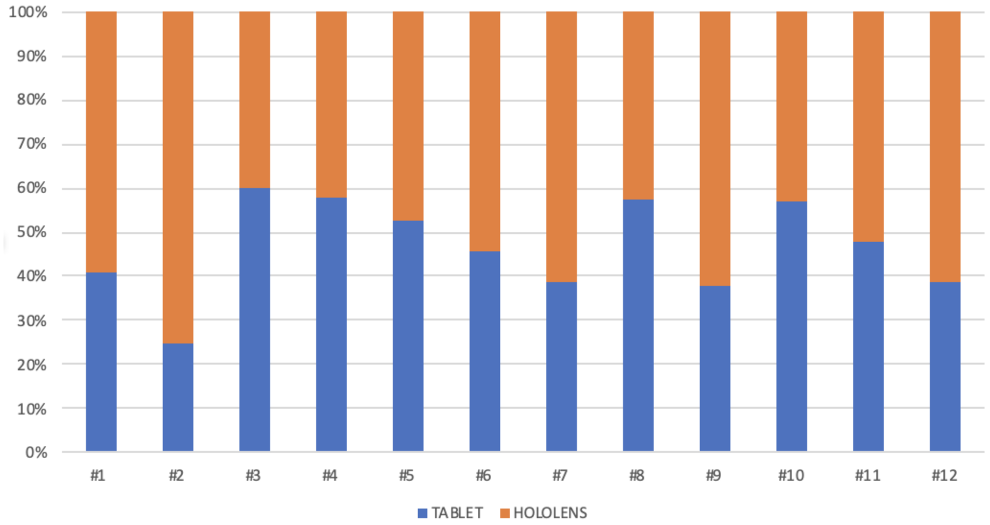


**Figure 5.6:** Tablet and HoloLens ratio regarding the task completion times per subject.

The figure 5.7 illustrates the relation of the mean operation times per task. Furthermore, it also contains a comparison between tablet and HoloLens for each task.

It is visible that the mean times for each operation are dependent on the task which is currently performed. For example, the percentage of translation on the tablet is nearly the half when executing task 1 (house) compared to task 2 (cactus) because the cactus consists of three sub-objects which have to be translated accordingly compared to two sub-objects for the house.

Nevertheless, the focus of the diagram in figure 5.7 is not on comparing the tasks which each other but to compare the tablet and HoloLens scenarios concerning a task. For instance, the percentage of the rotation operation in task 3 (wine glass) is higher

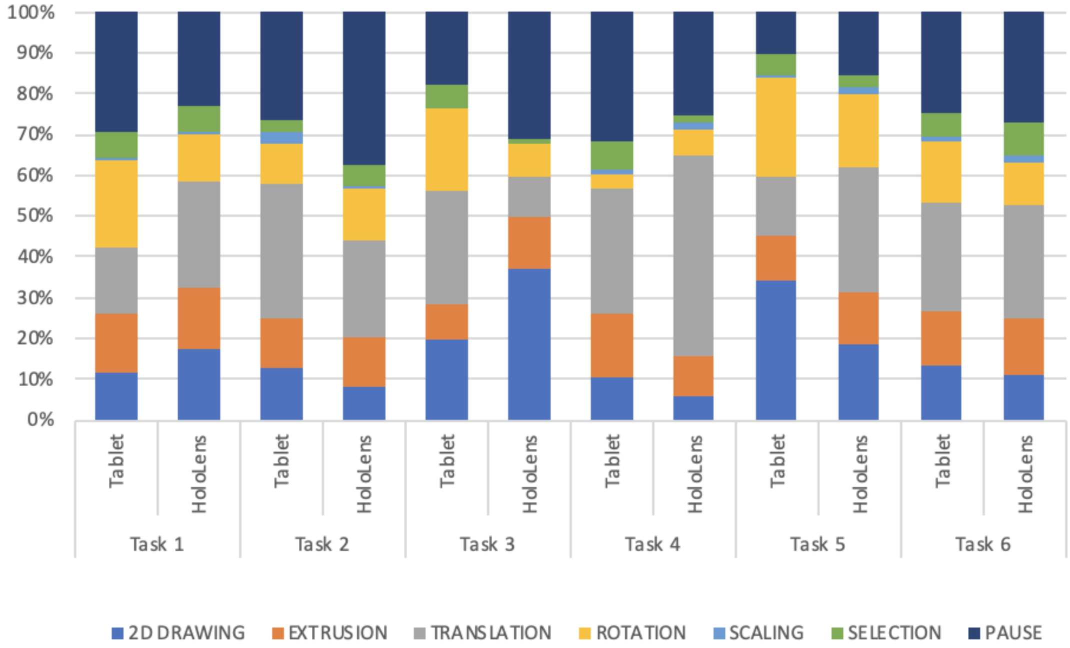


**Figure 5.7:** Mean operation times (%) per task and with comparison of tablet and HoloLens.

in the tablet scenario. The reason is assumed to be the visualization scenario. When the subjects have built the wine glass by drawing the silhouette and extrude it along a circle, they viewed the glass from below in the tablet scenario. To see the glass sidelong, they had to rotate it whereas, with the HoloLens, the participants already saw the full result during and after extrusion and therefore needed less rotation.

To show the mean operation times independently from the tasks, a second diagram is presented in figure 5.8. The diagram in figure 5.8 directly compares the mean operation times of the two visualization scenarios (tablet and HoloLens). The relation of the operation times related to the categories 2D drawing, extrusion, and selection are almost the same which means that the visualization scenario does not have any impact on the operation times of these operations.

Regarding the operation translation, there is a small difference between planar and stereoscopic visualization. The percentage of translation is higher in the HoloLens scenario. It is assumed that participants in the HoloLens scenario translate more because of the better perspective whereas participants in the tablet scenario are very limited concerning the depth perception even though the planar visualization is perspective (and not orthogonal!). Consequently, the subjects are less translating in z-direction (depth) when working exclusively with the planar visualization.

When considering rotation, it is reasonable that the operation time for rotation is higher for the tablet scenario because of the perspective of the planar visualization. The users want to see their created objects from different viewing angles. The stereoscopic visualization offers a better spatial perception of the created object and users can move

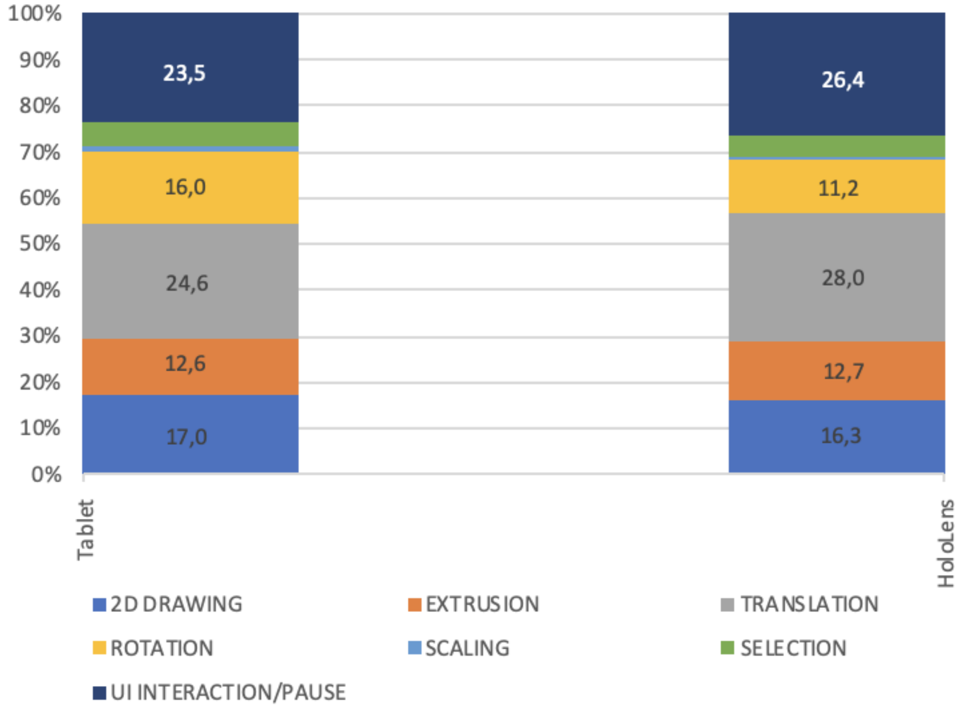


**Figure 5.8:** Mean operation times (%) presented int the tablet and the HoloLens scenarios.

their head to change the perspective immediately, and in an intuitive way and without UI interaction whereas in the tablet scenario, participants have to rotate the object to view it from different viewing angles.

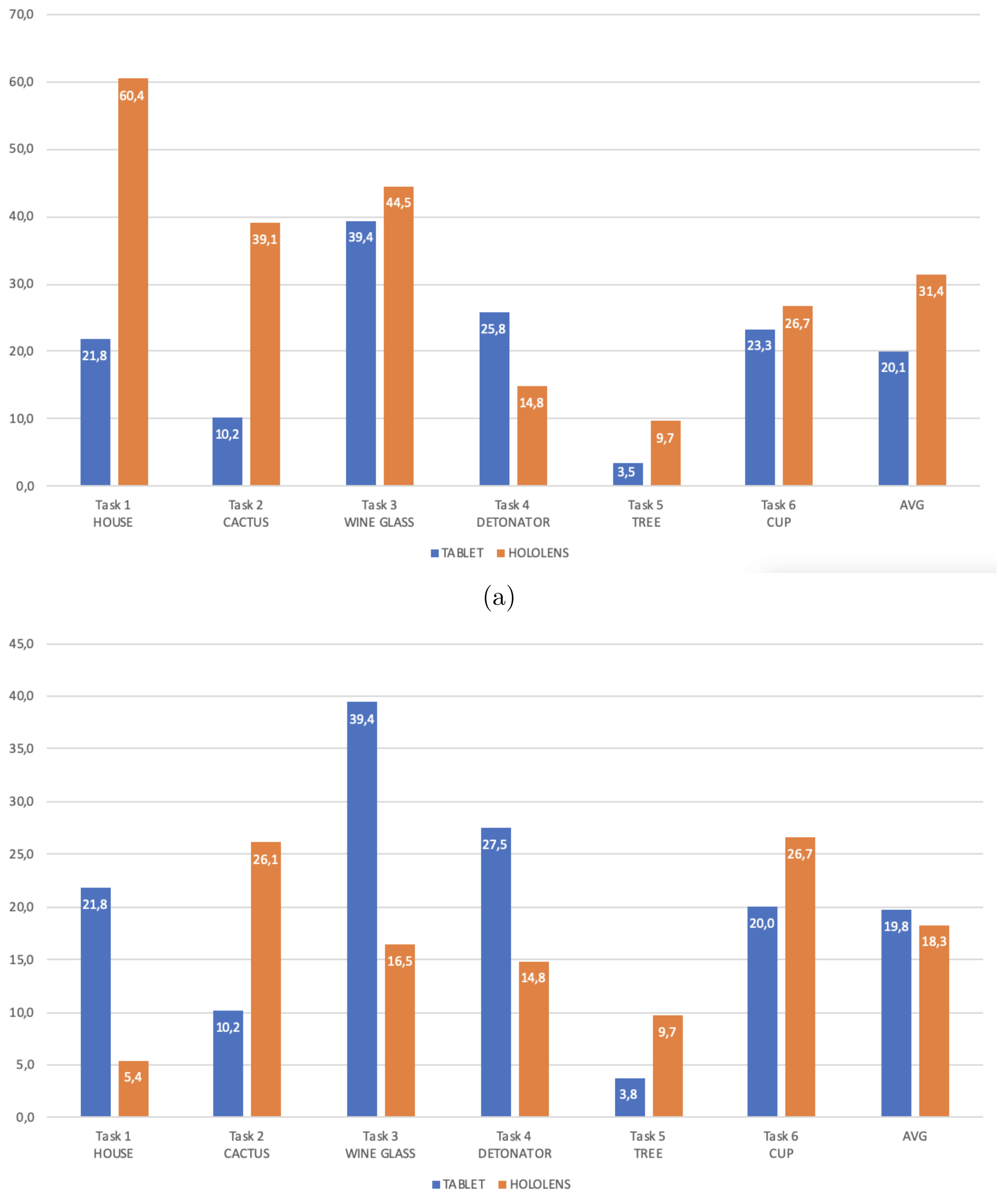
As already mentioned in the last section, the pause time is higher in the HoloLens scenario. Nonetheless, the reason for this tendency is not distinct because there are two possible reasons which have been observed during the study. The first reason for the higher pause time on the HoloLens scenario is the increased viewing of the stereoscopic visualization. The second reason is that participants need more time for changing the focus from the stereoscopic visualization to the tablet-based user interface because the stereoscopic visualization is partially occluding the user interface.

### 5.4.2 Error Rate

An error in the context of this prototype is defined to be either an undo, redo, or a delete operation. Hence, if the participant presses the undo, redo or delete button, an appropriate error log is added to the log file and afterwards counted as an error.

The diagrams in figure 5.9 illustrate the relation between the total number of errors (of all participants) per task and the total number of nodes (of all participants) per task regarding the tablet and the HoloLens scenario. It also shows the average values of the overall error rate (all tasks combined).

The diagram (a) in figure 5.9 indicates that the error rate is 11% higher on average



(a)

(b)

**Figure 5.9:** Error rate (%) in relation to nodes per task. The diagram in (a) shows the error rate with outlying and (b) without outlying participant.

when the participants are completing the tasks in the HoloLens scenario. The great disparity between the HoloLens and tablet scenario is caused by an outlier in the user study. The outlying participant was part of the second test group, which completed the first three tasks in the HoloLens scenario.

During the test sessions with this outlying participant, it was conspicuous that the participant was using the undo and redo feature a lot more than the other participants on average whereas other participants tended to immediately delete the whole object

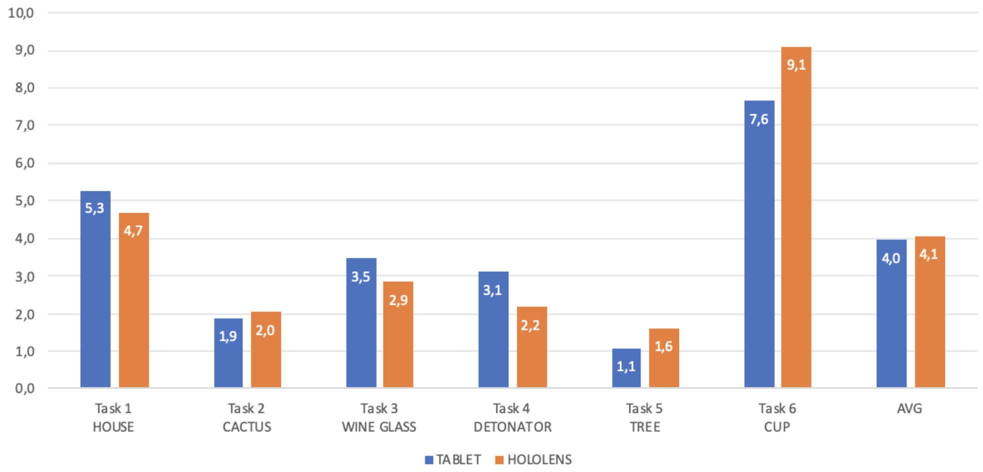


**Figure 5.10:** Delete object operations (%) in relation to nodes per task presented for the tablet and the HoloLens scenarios.

(which is counted as one single error) instead of undoing the incorrect parts of the object.

When the undo feature is used, the undo button was very often pressed several times in succession because the participant wanted to undo the whole object instead of deleting it at once which results in multiple errors (undo button is usually pressed several times) instead of one error (delete button is usually pressed only one time).

Therefore, the diagram (a) in figure 5.9 presents the error rate with the outlying participant who was using the undo operation for undoing the whole object instead of using the delete button because of personal preference. The diagram (b) in figure 5.9 presents the error rate without the outlying participant.

It is visible that the result of both diagrams in figure 5.9 is different because of the outlier. To filter the disruptive factor (some participants are using undo and redo a lot more than others) which results from combining delete, undo and redo operations for calculating the error rate, a third diagram was created which is shown in figure 5.10.

Regarding the diagram 5.10, an error is now considered to be an object deletion only. Hence, if the participant pressed the delete button, he had made an error. All participants were using the delete button in the same manner, and even the outlying participant was deleting the object after undoing all the steps he had made.

For this reason, the diagram in figure 5.10 evaluates the error rate of the tablet and the HoloLens scenario.

The error rate, shown in figure 5.10, of the tablet scenario is almost the same as the error rate when completing the tasks in the HoloLens scenario. The error rate of the HoloLens scenario is 0.1% on average higher than the error rate of the tablet scenario, which is a minor difference. Hence, the used visualization modality does not have a major impact on the error rate. When viewing the error rates (%) per task, no correlation can be found between the error rate and the corresponding task complexity.

**Table 5.3:** School grades and the corresponding description for evaluating precision of result objects.

| Grade | Description |
|---|---|
| 1 | all objects connected, correct placement of sub-objects, very similar to sample object |
| 2 | objects connected, sub-objects slightly displaced, shape differs slightly, similar to sample object |
| 3 | objects almost connected, moderate displacement of sub-objects, shape differs, dimensions of sub-objects unproportional, similar to sample object |
| 4 | objects clearly not connected, extensive displacement of sub-objects, shape of sub-objects look different, barely similar to sample object |
| 5 | objects not connected, large gaps, result not similar to sample object |

Consequently, the hypothesis H2 from section 5.2, which states that participants are making less mistakes when a stereoscopic visualization is available, is therefore disproved.

### 5.4.3 Precision Measurement

Regarding the precision of the resulting user-generated objects, a quantitative evaluation is not reasonable in case of this user study because even if the participants are prompted to rebuild the provided sample objects, it emerged that each of them has an individual view of the object. The participants recreate them in their way, and because the system is laid out for sketching 3D objects, high precision is tough to achieve. A quantitative measurement (for instance, with the *MeshLab* comparison functionality) is therefore not useful. Consequently, the resulting objects are rated with school grades. The grades have been assigned using the pattern presented in table 5.3.

The grades for the precision of the result objects are illustrated in figure 5.11. It clearly emerges that the precision grade is better when participants are using the HoloLens for stereoscopic visualization.

Especially, if a task requires many translations and rotations, it is more difficult for the participants to score a good grade in precision because the planar visualization is limited again in its depth presentation which is why the sub-objects are very often not connected.

The diagram in figure 5.11 also shows the average grade for the tablet scenario (average 2.5) and the HoloLens scenario (average 1.7). As a result, HoloLens users have finished with a grade better on average compared to participants in the tablet scenario.

Consequently, the hypothesis H3 from section 5.2, which states that participants are creating and reconstructing a given 3D object more precisely when a stereoscopic visualization is available, is therefore confirmed.

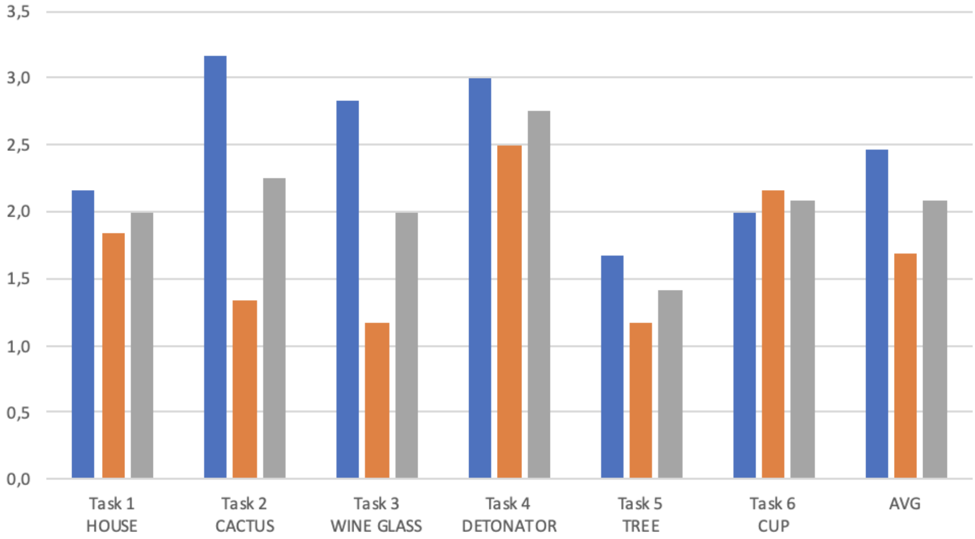


**Figure 5.11:** Precision is evaluated with school grades and per task.

### 5.4.4 NASA-TLX Questionnaire

After the participants have completed all task sessions, they will be asked to fill in the NASA-TLX[33] questionnaire sheet, which focuses on the workload for the user during interacting with the system.

The NASA-TLX is used for qualitative evaluation of the user study and consists of six categories which are mental demand (Was the task simple or complex?), physical demand (Was the task slack or strenuous?), temporal demand (Were the tasks paced slow or rapid?), overall performance (How successful was the participant?), effort (How hard did the participant work to achieve his level of performance?) and frustration level (How stressed, irritated, annoyed or content and relaxed was the participant?). For each category, the participant has to allocate a value from 0 to 20, whereas 0 is low or bad, and 20 is high or good.

The results of the NASA-TLX questionnaire are listed in figure 5.12 and shows the results for each participant and also the calculated average.

As shown in the diagram, the results for mental demand and effort are fluctuating around 11 on average, which means that the fulfillment of the tasks has been moderate, not too easy and not too complicated. The physical demand is four on average and therefore respectively low but nevertheless a meaningful result because all the participants were sitting during the task sessions and at most moving the upper part of the body to view the stereoscopic object. The temporal demand is five on average and low because there were no limits regarding time. Participants have chosen higher values if they were stressed because of other appointments. The overall performance is approximately 12 on average. Hence, participants had to invest a moderate effort to achieve their goal. This value results from the complexity level of the extrusion modality, which was not always easy for the participants to understand and apply. Resulting from this is a frustration level of 8 on average.

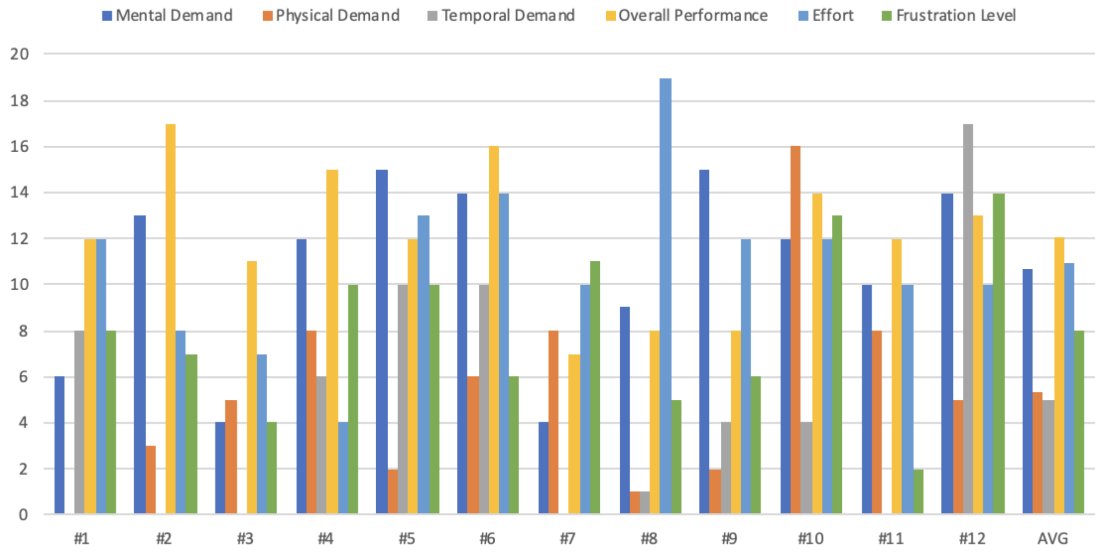


**Figure 5.12:** Evaluation

### 5.4.5 Feedback of the Participants

After each user study session, the participants could exchange their thoughts about the system.

#### Extrusion

The extrusion modality for creating objects was not trivial for every participant. Three participants had problems to understand extrusion and to complete the tasks in a meaningful time. Three participants have understood extrusion very fast and good.

#### Tablet Position

There were also different preferences regarding the position of the tablet. Four out of twelve participants preferred to have the tablet lay down straight on the table, whereas the rest preferred to have the tablet in a standing position. In this way, they can look down onto the stereoscopic visualization (bird´s eye view) and have a different perspective with the planar visualization on the tablet at the same time.

#### Which visualization scenario was easier for working with 3D objects and why?

Because the NASA-TLX questionnaire evaluates the whole system and not the comparison of the tablet and HoloLens scenario, in particular, the participants have also been asked which scenario they have enjoyed more to work with, which one has been easier to work with and why.

Three participants of twelve answered that they like the tablet scenario more because they felt irritated with the stereoscopic visualization directly placed on the tablet surface. The reason is that the stereoscopic visualization was placed in their working space where they were drawing and interacting. They had trouble to see the user interface underneath the stereoscopic objects and changing the focus (where eyes are looking at) from the stereoscopic object to the user interface was also tricky for them.

One participant also commented that it is not beneficial that the lines are hidden when the object is not selected. The prototype was implemented in this way to avoid too many lines and to keep the drawing space clean. A disadvantage resulting from the hidden lines is the difficulty of estimating the proportions of an object.

Nine participants enjoyed completing the tasks with the help of the HoloLens. The main reason was the perspective of the stereoscopic visualization provided. The augmentation helped to estimate the dimensions and proportions of an object, and it was, therefore, easier for the participants to recreate the provides sample object. Furthermore, they could move their bodies to view the augmented object from a different angle.

Consequently, the hypothesis H4 from section 5.2, which states that it is easier for participants to work with 3D objects in 3D space when a stereoscopic visualization is available, is therefore confirmed.

## 5.5 Summary and Discussion

In order to recapitulate the process of the user study and its evaluation, the following steps have been realized to achieve the main objective, which is finding an answer to the following research question.

> How does AR-based stereoscopic visualization support the user in creating extrusion objects, and what are the advantages and disadvantages of it?

First of all, to find an answer to the research question, four hypotheses have been set up which assume that with stereoscopic visualization, users can create and reconstruct a given 3D object faster, more precisely, with less mistakes and that stereoscopic visualization facilitates creating and reconstructing a provided sample object. A user study has been prepared for evaluating these hypotheses, in which twelve participants have rebuilt six sample objects in two visualization scenarios (tablet scenario without the HoloLens and HoloLens scenario).

Finally, all of the results, the user study provided, are comprehensible, but not all of them fulfilled the expectations, which makes them interesting.

The first hypothesis stated that participants are faster when the stereoscopic visualization is available, which has been disproved. The results have shown that the participants have been slower on average when wearing the HoloLens. On the one hand, participants needed more time to switch their focus back to the non-stereoscopic user interface after viewing the augmentation because the augmentation was occluding their workspace and on the other hand, participants spent more time on viewing the created objects from different viewing angles. The analysis of the mean operation times has also shown that the participants have spent more time in translating and less time in rotation operations when wearing the HoloLens.

The second hypothesis stated that participants are making fewer mistakes with the help of a stereoscopic visualization. Nevertheless, the result indicates that there is no major impact on the error rate when performing the tasks with or without the stereoscopic visualization because the error rates are almost the same in both visualization variants. This result is comprehensible because the mistakes are mostly made in conjunction with drawing and extruding operations which are equal in and independent from both variants. Hence, the hypothesis has been disproved.

The third hypothesis stated that participants are reconstructing the sample objects more precisely, which has been confirmed. Objects, which were created with the HoloLens, were graded one grade better on average than objects which were created solely with the planar visualization. The main reason therefor was the limited depth presentation of the planar visualization, and it was, therefore, difficult for participants to connect all sub-objects.

The fourth hypothesis stated that it is easier for participants to work with 3D objects in 3D space when a stereoscopic visualization is available, which has been confirmed. The usage of a stereoscopic visualization facilitates working with 3D objects in 3D space because the augmentation offers a better spatial perception of the created objects which eases rotating and translating sub-objects into the right position. Furthermore, the estimation of dimensions and proportions of an object was easier for the users.

## Chapter 6

# Conclusion and Future Work

The main objective was to find out if a stereoscopic visualization is supporting the user in creating objects and also to find advantages and disadvantages when working with stereoscopic visualization.

A prototype has been developed, which provides a user interface for sketching objects with extrusion, manipulating objects and also offers a planar and stereoscopic visualization. A methodical approach has been organized and realized step by step, which included concept design, implementation, and evaluation. In the course of concept design, the structure and appearance of the user interface and user interaction has been drafted for the prototype. In the implementation step, the designed prototype has been developed, which has been documented in chapter 4. The last step was to evaluate the resulting prototype with the help of a user study to find results that may answer the research question. The evaluation includes a quantitative and qualitative evaluation part and has been documented in chapter 5.

## 6.1 Final Results

To finally answer the research question, AR-based stereoscopic visualization supports the user in creating and reconstructing 3D objects more precisely. Compared to the planar visualization, stereoscopic visualization offers a better depth perception. It also facilitates working with 3D objects in 3D space because the augmentation offers a better spatial perception of the created objects which eases rotating and translating sub-objects into the right position. The augmentation also helped to estimate the dimensions and proportions of an object.

Nevertheless, it does not reduce the time spent on rebuilding objects. A disadvantage resulting from the larger time period when working with the HoloLens is that the augmentation is partly occluding the user interface on the tablet which makes it more difficult for the user to interact with the user interface. It was also difficult for some participants to change their focus from the augmentation back to non-stereoscopic visualization. An advantage, resulting from the larger time period is that users are taking more time to view the stereoscopic visualization from different viewing angles to get a better impression of the object.

Regarding the error rates when working solely with the planar visualization or with stereoscopic visualization, no major impact of one of the visualization forms was detected because the error rates have been very similar.

## 6.2 Challenges

During development, challenges occurred regarding the extrusion algorithm and the communication between the tablet and the HoloLens. Also, the evaluation and comparison of both visualization variants were challenging.

### 6.2.1 Extrusion

The implementation of the extrusion algorithm was very time-consuming because the extrusion of a custom 2D shape should work in real-time, along a custom 2D path and nevertheless be executed fluently without stuttering which means that the extrusion algorithm had to proceed with good performance.

### 6.2.2 Communication between Surface and HoloLens

The development platform for the Microsoft HoloLens is the Universal Windows Platform whereas the Microsoft Surface application is built as a Windows Standalone application. The different platforms of these two applications complicated the communication establishment because the used Web Socket library was not supported for UWP applications. Hence, a wrapper class was necessary for replacing the WebSocket class and its methods with another UWP compatible WebSocket class.

### 6.2.3 Comparison of Planar and Stereoscopic Visualization

The comparison between the use of the Microsoft Surface and the planar visualization of 3D objects and the use of the Microsoft HoloLens and the stereoscopic visualization was challenging because it was challenging to find a common basis on which the two variants could be compared with each other. All six tasks from the user study are different, which is why they can hardly be compared with each other. Also, each of the participants is an individual, whereas one is more experienced, one is more ambitious, and one is slower. Therefore, the participants have been split into two groups, and the order of the visualization variant was counterbalanced. One group started with the tablet scenario and the second group with the HoloLens scenario. A training session has also been included in the user study to reduce the learning effect.

## 6.3 Improvements and Future Developments

During the development of the prototype and the user study, a few suggestions for improvements have appeared which could be realized in future developments.

1. The stereoscopic visualization, when placed on the tablet, is partly occluding the user interface and therefore disturbing the user during the working process. The placement of the stereoscopic visualization may be more effective when placed alongside the tablet or using the stereoscopic visualization only when transforming

the object. So, an *activate* functionality could be added to let the user decide if he wants to use the stereoscopic visualization.

2. Selection of multiple objects would be helpful to rotate the whole object at once.
3. At the moment, the extrusion path is placed at the first point of the 2D shape. This circumstance causes compression or stretching of the 3D object when its 2D shape is extruded along a curve. Therefore, the extrusion path should start in the center of the 2D shape to create a consistent compression and stretching along curves.
4. The planar visualization of the 3D space could be improved by providing the user the possibility to navigate through the 3D space. A possible solution for navigating in 3D space with solely using a pen is shown in figure.
5. Maybe the object transformation gizmo used in the prototype is not optimal when stereoscopic visualization is active. The usage of sliders for all forms of transformations (translation, rotation, and scaling) may be better because the user interface would look more consistent and the placement of sliders at the side of the user interface may enhance working with the stereoscopic visualization because they may not be occluded too much by the stereoscopic augmentation.

# Appendix A

# Technical Details

## A.1 Screenshots

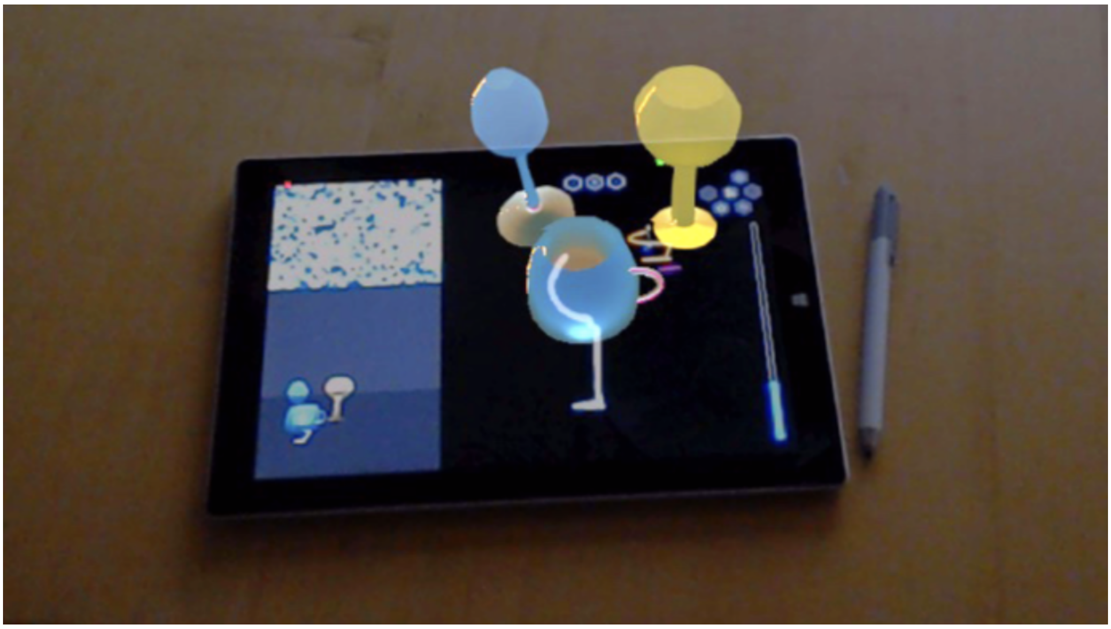


**Figure A.1:** Live capture of the stereoscopic visualization taken with the HoloLens camera.

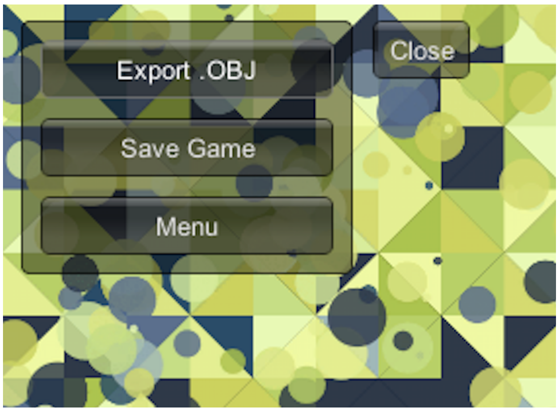


**Figure A.2:** Panel for saving and exporting the object and for returning to the menu panel.

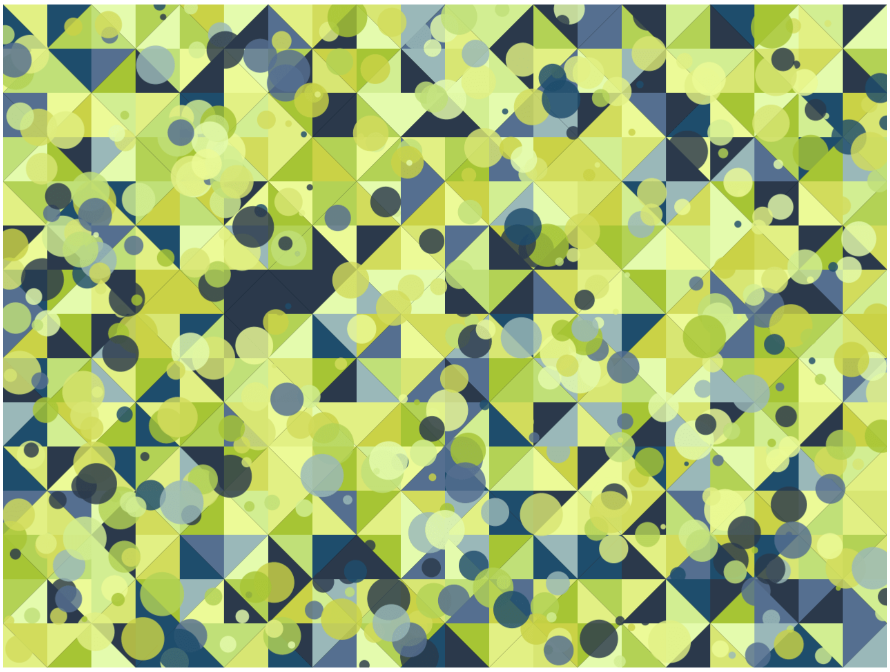


**Figure A.3:** The Image Target motive [34] used in the prototype.

# Appendix B

# Content of the DVD

Format: DVD+R, Double Layer, ISO9660-Format

## B.1 PDF-Files

Path: /

| | |
|---|---|
| MasterThesis.pdf | Master's Thesis |

## B.2 Project Data

### B.2.1 Microsoft Surface Application

Path: /implementation

| | |
|---|---|
| Demo.mp4 | Demo of Prototype |

Path: /implementation/surface_application

| | |
|---|---|
| MASurface.zip | Unity3D Project for Microsoft Surface |
| BuildSettings.png | Build Settings for Surface Application |

### B.2.2 Microsoft HoloLens Application

Path: /implementation/hololens_application

| | |
|---|---|
| MAHoloLens.zip | Unity3D Project for Microsoft HoloLens |
| BuildSettings.png | Build Settings for HoloLens Application |
| DeploySettings.png | Deployment Settings of VS Solution |

## B.3 User Study

Path: /evaluation/userstudy

| | |
|---|---|
| Evaluation.xlsx | Evaluation of User Study |
| OperationTimes.xlsx | Evalutaion of Operation Times |

Path: /participants/**/

| | |
|---|---|
| [participant]/logfiles/**.txt | Log Files of Participant |
| [participant]/result_objects/**.obj | Result Objects |
| [participant]/videos/**.mov | Video Recordings of Tasks |
| [participant]/**.xlsx | Data Collection of Participant |

Path: /questionnaire

| | |
|---|---|
| NASA-TLX.pdf | NASA-TLX Questionnaire in German |

Path: /template_objects

| | |
|---|---|
| main_tasks/task[1-6].* | Images of the Main Sample Objects |
| training_tasks/*.png | Images of the Training Sample Objects |

## B.4 References

Path: /references

| | |
|---|---|
| literature/[reference_title].pdf | Files of Referenced Literature |
| online/[reference_title].pdf | Files of Referenced Online Sources |

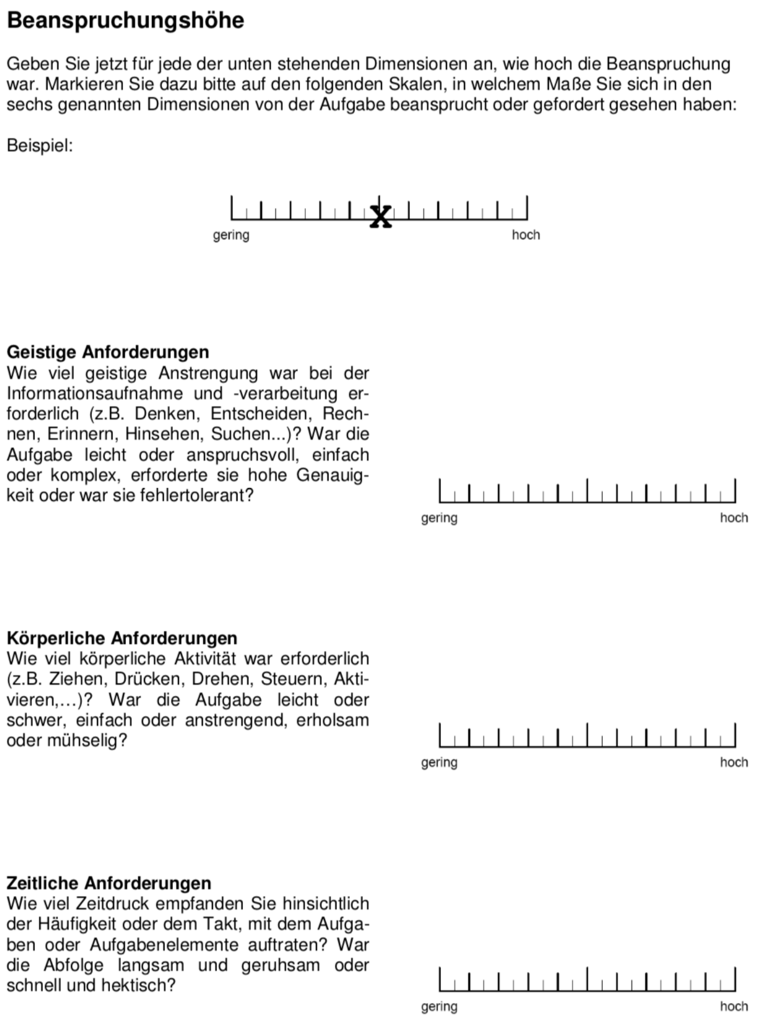
## B.5 Miscellaneous

Path: /images

| | |
|---|---|
| related_work/**.pdf | Images of Related Work Chapter |
| conceptual_design/**.pdf | Images of Conceptual Design Chapter |
| technical_impl/**.pdf | Images of Implementation Chapter |
| evaluation/**.pdf | Images of Evaluation Chapter |

# Appendix C

# Questionnaire



**Beanspruchungshöhe**

Geben Sie jetzt für jede der unten stehenden Dimensionen an, wie hoch die Beanspruchung war. Markieren Sie dazu bitte auf den folgenden Skalen, in welchem Maße Sie sich in den sechs genannten Dimensionen von der Aufgabe beansprucht oder gefordert gesehen haben:

Beispiel:

X

gering hoch

**Geistige Anforderungen**
Wie viel geistige Anstrengung war bei der Informationsaufnahme und -verarbeitung erforderlich (z.B. Denken, Entscheiden, Rechnen, Erinnern, Hinsehen, Suchen...)? War die Aufgabe leicht oder anspruchsvoll, einfach oder komplex, erforderte sie hohe Genauigkeit oder war sie fehlertolerant?

gering hoch

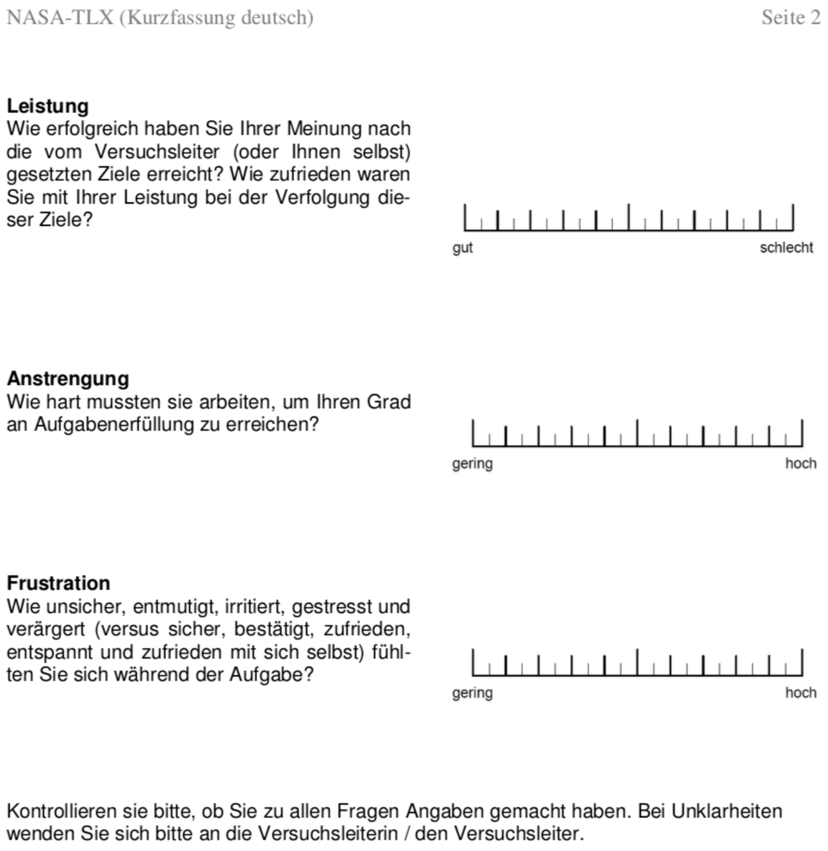
**Körperliche Anforderungen**
Wie viel körperliche Aktivität war erforderlich (z.B. Ziehen, Drücken, Drehen, Steuern, Aktivieren,…)? War die Aufgabe leicht oder schwer, einfach oder anstrengend, erholsam oder mühselig?

gering hoch

**Zeitliche Anforderungen**
Wie viel Zeitdruck empfanden Sie hinsichtlich der Häufigkeit oder dem Takt, mit dem Aufgaben oder Aufgabenelemente auftraten? War die Abfolge langsam und geruhsam oder schnell und hektisch?

gering hoch

**Figure C.1:** NASA-TLX Page 1



NASA-TLX (Kurzfassung deutsch) Seite 2

**Leistung**
Wie erfolgreich haben Sie Ihrer Meinung nach die vom Versuchsleiter (oder Ihnen selbst) gesetzten Ziele erreicht? Wie zufrieden waren Sie mit Ihrer Leistung bei der Verfolgung dieser Ziele?

gut schlecht

**Anstrengung**
Wie hart mussten sie arbeiten, um Ihren Grad an Aufgabenerfüllung zu erreichen?

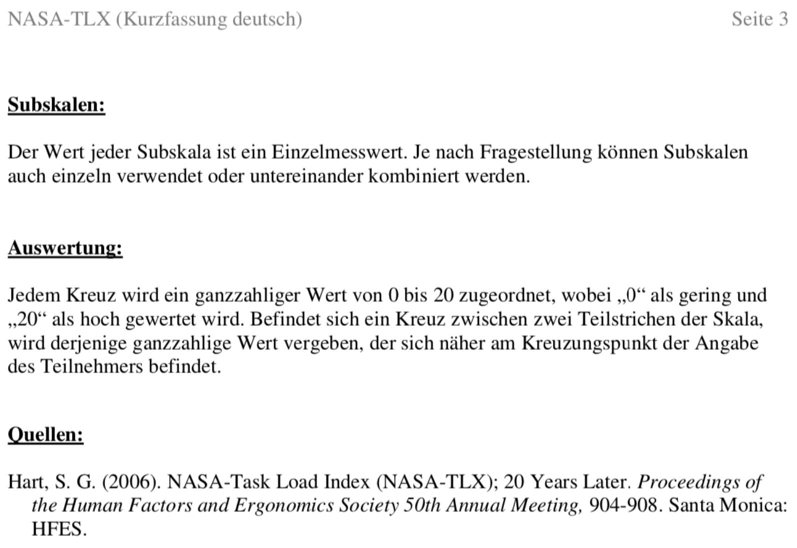
gering hoch

**Frustration**
Wie unsicher, entmutigt, irritiert, gestresst und verärgert (versus sicher, bestätigt, zufrieden, entspannt und zufrieden mit sich selbst) fühlten Sie sich während der Aufgabe?

gering hoch

Kontrollieren sie bitte, ob Sie zu allen Fragen Angaben gemacht haben. Bei Unklarheiten wenden Sie sich bitte an die Versuchsleiterin / den Versuchsleiter.

**Figure C.2:** NASA-TLX Page 2



NASA-TLX (Kurzfassung deutsch) Seite 3

**Subskalen:**

Der Wert jeder Subskala ist ein Einzelmesswert. Je nach Fragestellung können Subskalen auch einzeln verwendet oder untereinander kombiniert werden.

**Auswertung:**

Jedem Kreuz wird ein ganzzahliger Wert von 0 bis 20 zugeordnet, wobei „0“ als gering und „20“ als hoch gewertet wird. Befindet sich ein Kreuz zwischen zwei Teilstrichen der Skala, wird derjenige ganzzahlige Wert vergeben, der sich näher am Kreuzungspunkt der Angabe des Teilnehmers befindet.

**Quellen:**

Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting,* 904-908. Santa Monica: HFES.

**Figure C.3:** NASA-TLX Page 3

# References

## Literature

[1] Thomas Auer, Axel Pinz, and Michael Gervautz. "Tracking In A Multi-User Augmented Reality System". In: *Proceedings of the IASTED International Conference on Computer Graphics and Imaging.* Halifax, Canada, 1998, pp. 249–252 (cit. on p. 9).

[2] Ronald Tadao Azuma. "A survey of augmented reality". *Presence: Teleoperators & Virtual Environments* 6.4 (1997), pp. 355–385 (cit. on p. 4).

[3] 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. Monterey, CA, USA: ACM, 2008, pp. 151–160 (cit. on pp. 4, 5, 17).

[4] M. Shane Banks et al. "Stereoscopy and the Human Visual System." *SMPTE motion imaging journal* Vol. 121/4 (2011), pp. 24–43 (cit. on pp. 8, 9).

[5] Mark Billinghurst, Adrian Clark, and Gun Lee. "A Survey of Augmented Reality". *Foundations and Trends® in Human-Computer Interaction* 8 (Jan. 2015), pp. 73–272 (cit. on p. 9).

[6] Doug A. Bowman et al. "An Introduction to 3D User Interface Design". *Presence: Teleoperators and Virtual Environments* 10.1 (Feb. 2001), pp. 96–108 (cit. on p. 24).

[7] Gerd Bruder, Frank Steinicke, and Wolfgang Stuerzlinger. "Touching the Void Revisited: Analyses of Touch Behavior on and above Tabletop Surfaces". In: *Proceedings of the 14th International Conference on Human-Computer Interaction (INTERACT).* Vol. 8117. Cape Town, South Africa, Sept. 2013, pp. 278–296 (cit. on p. 8).

[8] Gerd Bruder, Frank Steinicke, and Wolfgang Stürzlinger. "To Touch or Not to Touch? Comparing 2D Touch and 3D Mid-air Interaction on Stereoscopic Tabletop Surfaces". In: *Proceedings of the 1st Symposium on Spatial User Interaction.* SUI '13. Los Angeles, California, USA: ACM, 2013, pp. 9–16 (cit. on p. 16).

[9] Scheurich Douglas and Stuerzlinger Wolfgang. "A One-Handed Multi-touch Method for 3D Rotations". In: *14th International Conference on Human-Computer Interaction (INTERACT).* Cape Town, South Africa, Sept. 2013, pp. 56–69 (cit. on p. 25).

[10] Lynn Eggli et al. “Inferring 3D models from freehand sketches and constraints”. *Computer-Aided Design* 29.2 (1997). Solid Modelling, pp. 101–112. URL: http://www.sciencedirect.com/science/article/pii/S0010448596000395 (cit. on p. 5).

[11] Zi-Gang Fu et al. “Reconstruction of intersecting curved solids from 2D orthographic views”. *Computer-Aided Design* 42 (Sept. 2010), pp. 841–846 (cit. on p. 7).

[12] A. Fuhrmann, H. Loffelmann, and D. Schmalstieg. “Collaborative augmented reality: exploring dynamical systems”. In: *Proceedings of the IEEE Conference on Visualization ’97.* Oct. 1997, pp. 459–462 (cit. on pp. 9, 10, 17).

[13] Islam Gharib. “Design of sketch-based interface to enhance creativity in conceptual design”. *International Design Journal* 4 (Apr. 2014), pp. 261–276 (cit. on p. 5).

[14] Paul E. Haeberli. “ConMan: A Visual Programming Language for Interactive Graphics”. In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques.* SIGGRAPH ’88. New York, NY, USA: ACM, 1988, pp. 103–111 (cit. on pp. 6, 7, 17).

[15] Sandra Hart. “Nasa-task load index (Nasa-TLX); 20 years later”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting.* Vol. 50. Oct. 2006, pp. 904–908 (cit. on p. 49).

[16] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. “Teddy: A Sketching Interface for 3D Freeform Design”. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques.* SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 409–416 (cit. on pp. 5, 6, 17).

[17] Géry Casiez Joaquim A. Jorge Bruno R. De Araújo. “Mockup Builder: Direct 3D Modeling On and Above the Surface in a Continuous Interaction Space”. In: *GI ’12 Proceedings of Graphics Interface 2012.* Canadian Information Processing Society Toronto, Ont., Canada, 2012, pp. 173–180 (cit. on pp. 10, 14–17).

[18] Hannes Kaufmann and Dieter Schmalstieg. “Designing Immersive Virtual Reality for Geometry Education”. In: *Proceedings of the IEEE Virtual Reality Conference (VR 2006).* IEEE, Mar. 2006, pp. 51–58 (cit. on pp. 9, 10, 13, 14, 16, 17).

[19] Marc Lambooij, Wijnand Ijsselsteijn, and Ingrid Heynderickx. “Visual discomfort in stereoscopic displays: A review”. In: *Proceedings of SPIE 6490 : Stereoscopic Displays and Virtual Reality Systems XIV.* Vol. 6490. San Jose, California, United States, Jan. 2007, pp. 6490.01–6490.13 (cit. on p. 8).

[20] John McIntire, Paul Havig, and Eric Geiselman. “What is 3D good for? A review of human performance on stereoscopic 3D displays”. In: *Proceedings of the SPIE - The International Society for Optical Engineering.* Vol. 8383. May 2012, pp. 28–41 (cit. on p. 8).

[21] Paul Milgram and Fumio Kishino. “A taxonomy of mixed reality visual displays”. *IEICE Transactions on Information and Systems* 77.12 (1994), pp. 1321–1329 (cit. on p. 4).

[22] Gur Mosheiov. “Scheduling problems with a learning effect”. *European Journal of Operational Research* 132.3 (2001), pp. 687–693 (cit. on p. 47).

[23] Ji-Young Oh. “Desktop 3D Conceptual Design Systems”. PhD thesis. York University, Department of Computer Science, May 2005 (cit. on pp. 11, 50, 51).

[24] Ji-Young Oh, Wolfgang Stürzlinger, and John Danahy. “Comparing SESAME and Sketching on Paper for Conceptual 3D Design”. In: *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2005, pp. 81–87 (cit. on pp. 6, 11, 15, 17, 50, 51).

[25] Ji-Young Oh, Wolfgang Stürzlinger, and John Danahy. “SESAME: Towards Better 3D Conceptual Design Systems”. In: *Proceedings of the 6th Conference on Designing Interactive Systems*. DIS ’06. University Park, PA, USA: ACM, 2006, pp. 80–89 (cit. on pp. 6, 11–13, 15).

[26] Chris Shaw et al. “Decoupled Simulation in Virtual Reality with the MR Toolkit”. *ACM Transactions on Information Systems* 11.3 (July 1993), pp. 287–317 (cit. on p. 10).

[27] Simon Shum et al. “Solid reconstruction from orthographic opaque views using incremental extrusion”. *Computers & Graphics* 21 (Nov. 1997), pp. 787–800 (cit. on pp. 7, 8).

[28] Zsolt Szalavári and Michael Gervautz. “The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality”. *Computer Graphics Forum* 16 (Sept. 1997), pp. 335–346 (cit. on p. 9).

[29] Dimitar Valkov et al. “2D touching of 3D stereoscopic objects”. In: *Proceedings of the Conference on Human Factors in Computing Systems*. May 2011, pp. 1353–1362 (cit. on p. 8).

[30] Charlie Wang and Matthew Yuen. “Freeform extrusion by sketched input”. *Computers & Graphics* Vol. 27 (Apr. 2003), pp. 255–263 (cit. on pp. 5, 6).

[31] Stanislaw Zabramski and Wolfgang Stuerzlinger. “The Effect of Shape Properties on Ad-hoc Shape Replication with Mouse, Pen, and Touch Input”. In: *Proceeding of the 16th International Academic MindTrek Conference*. MindTrek ’12. Tampere, Finland: ACM, 2012, pp. 275–278 (cit. on p. 5).

[32] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. “SKETCH: An Interface for Sketching 3D Scenes”. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’96. New York, NY, USA: ACM, 1996, pp. 163–170 (cit. on pp. 5, 17).

## Online sources

[33] Interaction Design Group/Institut für Industrial Design/FB Ingenieurwissenschaften und Industriedesign/Hochschule Magdeburg-Stendal. *NASA-TLX (Kurzfassung deutsch)*. URL: http://interaction-design-group.de/toolbox/wp-content/uploads/2016/05/NASA-TLX.pdf (visited on 07/15/2019) (cit. on pp. 49, 60).

[34] Dario Mazzanti. *AR Image Target Generation (Vuforia) - Automatic generation of AR trackable targets.* URL: https://www.dariomazzanti.com/uncategorized/ar-image-target-generation-vuforia/ (visited on 07/15/2019) (cit. on pp. 43, 68).

# Check Final Print Size

— Check final print size! —

width = 100mm
height = 50mm

— Remove this page after printing! —