

Augmenting Security Systems

The Role of Augmented Reality in the Surveillance Industry

Johan Svedberg and Tobias Olsson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2018

MASTER THESIS



Augmenting Security Systems

The Role of Augmented Reality in the Surveillance Industry

Johan Svedberg and Tobias Olsson



LUND
UNIVERSITY

Augmenting Security Systems

The Role of Augmented Reality in the Surveillance Industry

Copyright © 2018 Johan Svedberg & Tobias Olsson

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)

Division: Ergonomics and Aerosol Technology

Supervisor: Günter Alce

Co-supervisors: Staffan Olsson and Ted Hartzell

Examiner: Joakim Eriksson

Abstract

As the focus of Augmented Reality (AR) shifts from consumer entertainment to industry appliances, there is a big need for exploratory studies to figure out what role they will play in the years to come.

This thesis aims to investigate how AR could be used to enhance the selling, planning and installation of a security system.

A prototype was developed to demonstrate some of the potential benefits of AR in the context. Finally, a user test was conducted on 24 people. The test evaluated both general usability and specific interaction models. Two versions were compared. The results suggested that our AR application was easy to use even for very inexperienced users. It also showed that even though users said they preferred to handle placement of objects themselves, the version in which objects followed the users peripheral vision performed better in all the tests.

Keywords: Augmented Reality, Microsoft HoloLens, Interaction, Security Systems, Surveillance

Sammanfattning

När användningsområdena för förstärkt verklighet (eng. Augmented Reality [AR]) nu börjar skifta från underhållning till industriella produkter, finns det ett stort behov av utforskande studier för att lista ut vilken roll de kommer att få under de kommande åren.

Detta examensarbete undersöker hur AR skulle kunna användas för att förbättra försäljning, planering och installation av ett säkerhetssystem.

En prototyp har utvecklats för att visa på några potentiella fördelar av AR i kontexten. Till sist utfördes en användarstudie på 24 personer. Studien utvärderade både generell användbarhet samt specifika interaktionsmodeller. Två versioner jämfördes. Resultaten visade att vår AR-applikation var enkel att använda, även för väldigt oerfarna användare. Det visade sig också att även om användarna sa att de föredrog att hantera placering av objekt själva, så fick versionen då objekten automatiskt följde efter i användarens synfält bättre resultat i samtliga test.

Keywords: Augmented Reality, Microsoft HoloLens, Interaction, Security Systems, Surveillance

Acknowledgements

We would like to thank Günter Alce, Staffan Olsson, and Ted Hartzell for the incredible support and all the lighthearted meetings throughout the thesis. Without you this would not have been possible. Additionally, we would like to thank all test participants. Thank you for giving us your precious time, and great feedback.

Last but not least, a big thank you to the Core Technologies Systems department, and the rest of Axis Communications, for the amazing support, and for making us feel like a part of the team.

¡ARriba, ARriba!

Contents

1	Introduction	7
1.1	Background	7
1.2	Purpose and goal	8
1.3	Scope and limiting factors	9
1.4	Axis Communications	9
1.5	Abbreviations	9
2	Theoretical and technical background	11
2.1	Augmented Reality	11
2.2	Virtual Reality	12
2.3	Usability evaluation	13
2.3.1	NASA Task Load Index	13
2.3.2	System Usability Scale	14
2.4	Software tools	14
2.4.1	Unity	14
2.4.2	Microsoft Visual Studio	14
2.4.3	Mixed Reality Toolkit	14
2.4.4	Urho3D and Xamarin	14
2.4.5	ARCore/ARKit	15
2.4.6	Git and Git Large File Storage	15
2.5	Microsoft HoloLens	16
2.5.1	Hardware	16
2.5.2	Using the HoloLens	16
3	Methodology	19
3.1	Process	19
3.1.1	Brainstorming	19
3.1.2	Initial development	19
3.1.3	Finalizing the application and User testing	21
3.2	Feedback and User-Centered Design	22

4	Application development	25
4.1	Ideas and requirements	25
4.1.1	Interacting with different camera models	26
4.1.2	Mapping the room and building a World in Miniature	26
4.1.3	Simulate a security system	27
4.1.4	Visualize what a camera can see	28
4.2	Interacting with the application	28
4.2.1	UI elements	29
4.2.2	Voice commands	31
5	Final evaluation	33
5.1	Purpose and goal	33
5.2	Participants	33
5.3	Test setup	36
5.4	Questionnaires	37
5.5	User test execution	38
5.6	Test results	39
5.6.1	SUS and NASA TLX	40
5.6.2	Interviews	42
6	Discussion	45
6.1	User study	45
6.2	Using AR to implement security systems	46
6.3	Interaction methods	47
6.3.1	Hold to drag	47
6.3.2	Tap to place	47
6.3.3	Toggle method	47
6.3.4	Two handed manipulation	48
6.3.5	Voice commands	48
6.3.6	Using UI elements	48
6.4	Future work	49
7	Conclusion	51
	Appendix A Personal information questionnaire	59
	Appendix B Modified SUS questionnaire	63
	Appendix C Interview Questions	67
	Appendix D Test scenario and user guide	69
	Appendix E Test plan	73
	Appendix F Information and consent	79
	Appendix G Tutorial and briefing	83

Chapter 1

Introduction

This introductory chapter will cover the background, purpose and goal, and scope of this thesis. It will also introduce some abbreviations that will be used throughout this thesis.

1.1 Background

In the surveillance industry, as in many other high-tech industries, an every-day problem might have a highly complex solution. Intrusion-detection and facial recognition, for instance, are problems that are very easy to understand, but requires performant systems that are complex to configure. This simplex-complex duality is often manifested in the relations between the customer, the project manager and the person installing the system. The project manager must understand the requirements of the customer, explain the possibilities and limitations of solutions to the customer, and relay details of the selected solution to the installer. The installer will have to be able to discuss adaptations of the selected solution with the project manager and to verify that the selected solution fulfills the customer's and the project manager's requirements.

The complex system configurations can cause problems for everyone involved in the implementation of a security system, and the planning and design of a large system takes a lot of man hours. If there is a risk that a potential collaboration or contract gets canceled, suppliers might be hesitant to even provide a proposition to the customer, due to the commitment in time and resources it takes to set up. Another problem that arises is that customers might not fully understand the complexity of a system, or the parameters which it relies upon. If a customer is not sure about what to expect from a product, it might harm both the confidence towards the system, as well as the chances of reaching an agreement with the supplier. Lastly, there might be installers that are reluctant to implement systems with certain functionality or complexity since they feel they do not have the knowledge or experience required. If something goes wrong in the installation phase, there is a big risk of having to redo the work, which costs both time and money.

One way to solve these problems could be using Augmented Reality (AR), overlaying virtual images and information onto the real world. After the initial wave of hype around AR and Virtual Reality (VR), with focus on consumer and entertainment products such as Pokemon GO [31] and a variety of VR headsets, the market has turned its eye towards businesses and industry applications. Telepresence, contextual information, and retail solutions has started to pop up everywhere. It can be used to guide less experienced users to complete complicated tasks, or help visualize something that cannot be done in the real world, like trying on clothes that are not available at a store. Figure 1.1 shows a woman trying on a virtual outfit in AR.

Could AR and VR be a good fit for tackling the problems facing the surveillance industry?



Figure 1.1: An example of an AR application in a store. Image taken from Shopify [41].

1.2 Purpose and goal

At the Core Technologies Systems (CTS) department at Axis Communications, there is an ongoing project called Installation Futura, exploring how the installation and planning of a security system could look like in the future. The idea behind this thesis is to explore what role AR and VR could play in the project, and in the installation process of an Axis camera system. The installation process ranges from the customers describing what problem they want solved, to the planning and actual installation of the cameras. The aim is to figure out what, if any, part AR and VR has to play in that process. This means visualizing complex problems and data; using other means than a traditional screen. For example, how do you show where a camera has to be placed to allow for the resolution required for facial recognition? Or demonstrate how different camera models perform in different light conditions?

In addition, we want to evaluate a couple of ways to handle complex interactions in AR. Can traditional metaphors such as the desktop environment be applied to AR, or is there a need for new solutions?

In the end, we hope to have produced one or more prototypes to showcase our findings, as well as a user study demonstrating how well they performed, while also comparing different ways of placing and handling holograms.

1.3 Scope and limiting factors

Even though we are experiencing a renaissance of sorts in the world of AR and VR, the currently available software and hardware leave a lot to be desired. Limited field of view (FOV), subpar tracking and uncomfortable hardware are some of the problems plaguing AR and VR still. Even though it is improving with incredible pace at the moment, these problems make their presence known. This limits many ideas to just that, ideas.

As always in these types of projects, time is a limiting factor. Two people can only do so much in the limited time-frame of the master thesis, and “killing one’s darlings”, the concept of limiting the scope, has to be enforced. In the 20 weeks, the prototype has to be developed, the user testing conducted, and the thesis itself written.

1.4 Axis Communications

In 1984, Axis Communications was founded in Lund, Sweden, by Martin Gren, Mikael Karlsson and Keith Bloodworth. The company started out as a manufacturer of print servers, and later on applied their innovative network technologies to create the world’s first network camera in 1996. Today, Axis Communications is the market leader in network video, always striving to explore and innovate new technology [2, 10].

1.5 Abbreviations

The following list contains some often used abbreviations, complete with a quick explanation.

AR - Augmented Reality

A system that displays virtual objects that appear to coexist in the same space as the real world [5].

FOV - Field of View

How much of the observable world a person sees at a given time. Often used to indicate restriction, when wearing goggles etc [3].

UI - User Interface

The means with which a person controls a software application or hardware device [9].

HMD - Head Mounted Display

A display worn on the head, allowing for free head motion, and in some cases full body mobility [36].

SUS - System Usability Scale

A subjective questionnaire, used to assess the usability of a system [8].

NASA-TLX - NASA Task Load Index

A subjective assessment tool that can be used to measure an estimate of the workload of a task or a system [18].

MR - Mixed Reality

A system that presents real world and virtual world objects together within a single display [30].

VR - Virtual Reality

A computer system which can generate a man-made world, in which the user can immerse, roam, and operate objects [46].

WiM - World in Miniature

A miniature, virtual, copy of the environment. Can be used for interaction and overview [42].

Chapter 2

Theoretical and technical background

In order to better understand the thesis, this chapter gives some important context and background information on terminology and key concepts.

2.1 Augmented Reality

Ronald T. Azuma defines AR as “An ... system [that] supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world”, as well as satisfying three main characteristics [5]:

- Combines real and virtual objects in a real environment.
- Runs interactively, and in real time.
- Registers (aligns) real and virtual objects with each other.

This can be accomplished in many ways, recording the real environment with a camera and overlaying the virtual objects as in Figure 2.1, or using a transparent display to show to virtual objects in the real environment. On the Milgram Reality-Virtuality continuum, seen in Figure 2.2, AR can be found close to the leftmost position, approaching the real or physical environment, and can be considered a subset of Mixed Reality (MR). According to Milgram, MR can be defined as an “environment ... in which real world and virtual world objects are presented together within a single display” [30].

Yet another distinction can be done by categorizing the AR devices by usage; handheld, stationary and head-mounted. Handheld devices include smartphones, tablets etc. Stationary devices can be computer or television-screens. Head-mounted displays (HMD) are gaining traction, but are still seldom seen in the hands (or on the heads) of consumers. Examples include the Microsoft HoloLens and the Meta 2 [22, 25].

A key feature of AR is knowledge about the physical reality. To achieve this, a marker based system can be used. As demands on mobility and usability increase, using markers



Figure 2.1: A televised NFL game, using AR to improve the viewing experience. Image taken from [12].

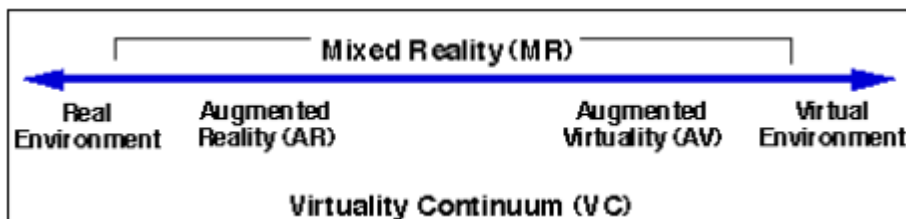


Figure 2.2: The Milgram Reality-Virtuality Continuum. Image taken from Milgram [30].

can be cumbersome. Instead, devices utilize modern sensors and sensor-fusion to map the environment in real time. The past couple of years has brought several devices to market with such solutions, most notably many smartphones now include very sophisticated sensors for use in AR applications. Lately, these devices have been using Simultaneous Localization and Mapping (SLAM) technology to realize the world around them [45].

2.2 Virtual Reality

Instead residing on the rightmost side of the Milgram RV continuum, Virtual Reality immerses the user completely in the virtual environment. This is often accomplished with a HMD, such as the HTC Vive [11].

Ning-Ning Zhou utilizes many different definitions of VR to come to the conclusion that they more often than not, encompasses both hardware and software used to create and interact with the virtual environment [46].

2.3 Usability evaluation

To gather user data and to evaluate the usability of our application, we used two well known evaluation questionnaires; NASA Task Load Index (TLX) and System Usability Scale (SUS).

2.3.1 NASA Task Load Index

NASA-TLX is a subjective assessment tool that can be used to measure the estimated workload of a task or a system. It was developed in the 1980's at NASA Ames Research Center by Sandra Hart [18]. NASA-TLX is split into two parts. In the first part the participant specifies how much workload was needed based on six different subscales. Each subscale is given a rating between 0-100. The six subscales and their definitions (quoted from Hart, 1988) are the following [18]:

Mental Demand: How much mental and perceptual activity was required (e.g, thinking, deciding, calculating, remembering, looking, searching etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Physical Demand: How much physical activity was required (e.g, pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

Temporal Demand: How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

Performance: How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

Effort: How hard did you have to work (mentally and physically) to accomplish your level of performance?

Frustration Level: How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

In the second part, the participant weights the different subscales against each other, deeming which of them had the most effect on the workload. The initial rating and the weights for each subscale are then compiled into a weighted workload rating. These weighted ratings are added together, which gives an overall workload score within the range of 0-100. A deeper explanation of how the overall workload score is calculated can be found in the Task Load Index paper [16]. Grier compiled over 1000 NASA-TLX scores to interpret how a certain TLX score stands amongst others of the same type [15]. She concludes that this data is a good start, but to decide if a workload is acceptable, one must not only look at the workload score, but also at a number of contextual variables.

2.3.2 System Usability Scale

SUS is another subjective assessment tool that primarily focuses on evaluating the usability of a system, product or application. It was developed by John Brooke in 1986 [8]. Compared to NASA-TLX, SUS is a much more lightweight tool, often referred to as a “quick and dirty” evaluation method. It consists of ten statements, where the users will fill out their level of agreement on a Likert scale, usually ranging between 1 to 5. The statements will alternate between “positive and negative” statements. The ratings are compiled using a formula, which will yield a result within a range of 0 to 100. A rating higher than 68 is usually considered an above average result and vice versa [38]. A more detailed interpretation of what a SUS rating describes is given by Bangor [6].

2.4 Software tools

This section will present the software that was used or considered important to the thesis.

2.4.1 Unity

Unity is a cross-platform game engine made by Unity Technologies, complete with a fully featured editor. The engine can run on 25+ platforms, with mobile, VR and AR support. Unity was the main development platform during this thesis.

2.4.2 Microsoft Visual Studio

Visual Studio is an integrated development environment (IDE), created by Microsoft. Its code editor was our primary external scripting tool in Unity and was also used to deploy our AR applications to the HoloLens.

2.4.3 Mixed Reality Toolkit

Mixed Reality started as an academic umbrella term, describing the many different ways to combine virtual and real objects, but in the last few years, the term has been adopted by Microsoft to denote their AR and VR platform [29]. The Mixed Reality Toolkit (MRTK) is “a collection of scripts and components intended to accelerate development of applications targeting Microsoft HoloLens and Windows Mixed Reality headsets.” We used the MRTK - Unity variant, which provides easy access to Microsoft’s Mixed Reality API from inside Unity, as well as some graphical assets and utility scripts, seen in Figure 2.3.

2.4.4 Urho3D and Xamarin

Xamarin is a Microsoft owned framework for cross-platform development. A developer working at Xamarin, Egor Bogatov, has implemented a way to work with the different AR-frameworks (Windows MR, ARCore and ARKit) using much of the same codebase [7]. This is enabled by Urho3D, a free, cross-platform 2D and 3D game engine.

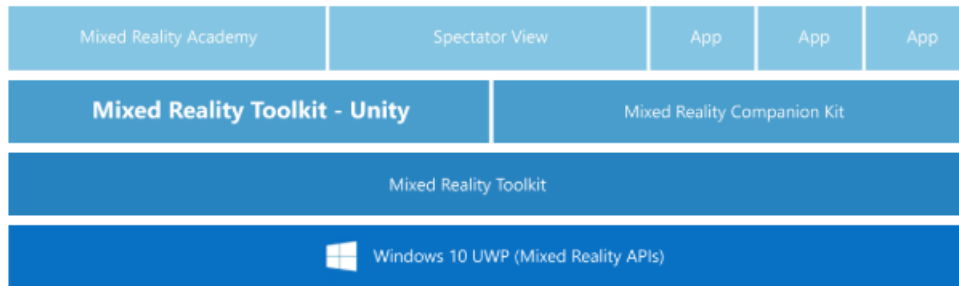


Figure 2.3: The many layers of HoloLens development using the MRTK. Image taken from Microsoft [26].

2.4.5 ARCore/ARKit

ARCore and ARKit are Google’s and Apple’s respective AR frameworks [14, 4]. They are similar in functionality, with the big difference being that ARKit only runs on Apple’s mobile operating system iOS, while ARCore only runs on Android.



Figure 2.4: An Apple iPad running an ARKit application. Image taken from Apple [4].

2.4.6 Git and Git Large File Storage

Git was created by Linus Torvalds in 2005, and is an open source version control system [1, 43]. In this project, Git is used to simplify collaboration between developers, to track code changes and to facilitate branching for parallel development. Git Large File Storage (LFS) is a Git extension which helps to keep track of changes to large files without having to store these in a local Git repository, but rather storing them at a remote site [13].

2.5 Microsoft HoloLens

The Microsoft HoloLens is the primary hardware used during the thesis. This section will give an introduction to its specifications and how it can be used and interacted with.

2.5.1 Hardware

The Microsoft HoloLens is, per Microsoft's own words, a "self-contained, holographic computer" [25]. A HMD, the HoloLens displays holograms in a real environment using a transparent screen. Since it is "self-contained", there is no need for external power or processing, enabling the user to walk around the space unobstructed.

It was released in Q1 2016, as a development edition meant for developers and companies. It has yet to be released as a fully fledged consumer product.

The HoloLens weighs about 579g, and has four "environment understanding cameras" enabling spatial understanding. The battery allows for 2-3 hours of active use, and the device is passively cooled [24].

2.5.2 Using the HoloLens

In order to interact with holograms, the HoloLens utilizes a couple of gestures. This section will describe them, and introduce some important concepts.

Gestures and the gesture frame

The gesture frame is the area in which the HoloLens can recognize interaction gestures. It is common that users believe that the gestures have to be right in front of the display, directly onto the holograms or within the FOV, but as illustrated in Figure 2.5, the gesture frame is a lot larger than that.

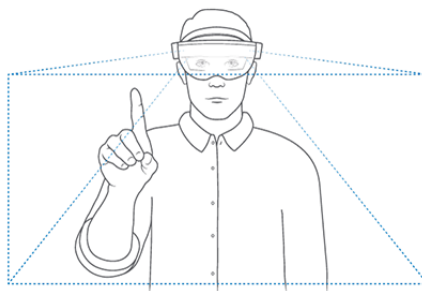


Figure 2.5: The HoloLens gesture frame. Image taken from Microsoft [27].

At this time, the HoloLens uses two different gestures to enable navigation and interaction with the holograms; Air Tap and Bloom (Figure 2.6 and 2.7).

Air tap is the basic click-method on the HoloLens. It is performed by forming a L-shape with the index finger and the thumb, and then "pinching" by moving the fingers

together. The HoloLens will perceive this as a click interaction. The two steps of the gesture is visualized in Figure 2.6.

The bloom gesture is a system-wide way to return to the HoloLens main menu. The gesture is visualized in Figure 2.7. This gesture is not meant to be used as an interaction gesture in ordinary HoloLens applications, since it will minimize the active application on use.

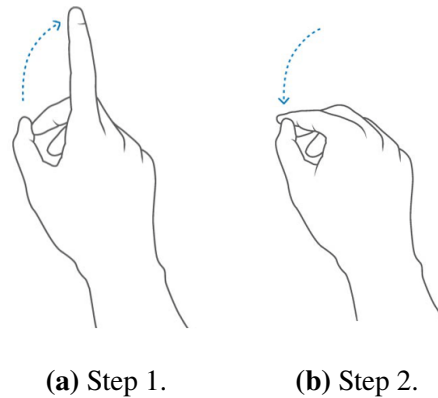


Figure 2.6: The Air Tap gesture. Image taken from Microsoft [27].

Hold to drag

Hold to drag is very similar to the drag and drop interaction of an ordinary computer mouse; keeping the “selection”-gesture active on an object while moving the hand will make the object follow the hand’s movement, as long as the hand is inside the gesture frame.

Tap to place

Tap to place is another way to move objects. By pressing on an object, the user is moved into a special placing mode, where the object in question is transported by the middle of

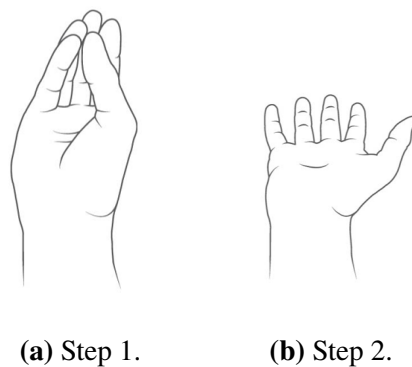


Figure 2.7: The Bloom gesture. Image taken from Microsoft [27].

the HoloLens' view, updating every frame in order to match the user's gaze. Pressing a second time will release the object and place it where the gaze was at the time. It is also possible to toggle spatial mapping awareness, enabling the object to snap to real objects or surfaces mapped by the HoloLens.

Toggle method

A commonly used way to switch between different object manipulations (like moving, rotating and scaling) is to implement a toggle functionality. Usually, there is a UI element with which a user can select the preferred manipulation method, and then the object will be affected accordingly when interacted with, by using the hold to drag method.

Two handed manipulation

Two handed manipulation recently got supported in the MRTK [34, 44]. By using the hold to drag method with both hands inside the gesture frame, a user has the possibility to use two handed manipulation. The interaction is similar to how one would interact with images on a phone or a tablet.

Voice commands

The HoloLens also supports voice input. For one, it is possible to utilize Cortana, Microsoft's virtual assistant via the HoloLens [23]. Apart from the standard functionality of Cortana, there are a few keywords that are specific to the HoloLens which can be found at Microsoft's mixed reality development page [28]. The HoloLens also supports speech recognition and voice input based on the Speech Recognition Grammar Specification [20] which is available on all standard Universal Windows applications [28]. This allow developers to add custom voice commands to their applications.

Chapter 3

Methodology

This chapter will describe the development process, and how user feedback was used to improve the application.

3.1 Process

The development process consisted of four main stages, which will be described in this section; *Brainstorming*, *Initial development*, *Finalizing the application* and *User testing*.

3.1.1 Brainstorming

Initially, many of our brainstorming sessions tackled the very important questions; What are we trying to solve? How are we going to solve it? Why are we trying to solve it? A potential system for future project planning and installation started to take place on a couple of whiteboards, for example the one in Figure 3.1.

Low fidelity prototypes were created to visualize our ideas, one example can be seen in Figure 3.2. Together with Axis, we agreed that building a model of the environment should be a major focus of the system. The model should be easy to interact with using all kinds of devices; phones, tablets, headsets, desktop computers etc. Another important part of the project should be to visualize security solutions and make it easier for everyone involved to understand them.

3.1.2 Initial development

Once the project started to take form, we quickly moved on to building some basic prototypes. At first, we experimented with many tools, like Xamarin, ARKit and Unity, as well as discussed what hardware we should use. How could we best model a room? Some

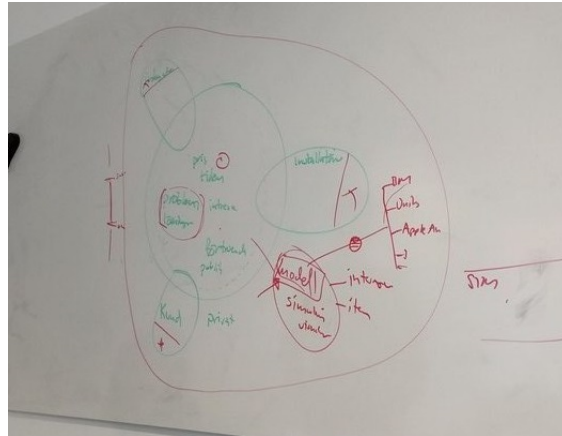


Figure 3.1: The results of an early brainstorm session.

initial ideas revolved around specialized hardware such as Structure, a 3D scanner that supports the scanning of an entire room [33]. We also thought a lot about how we could integrate AR and VR into a single solution. As the discussions continued, it became more and more clear what could be done with today’s technology, and ultimately we settled on using the Microsoft HoloLens together with Unity. The choice of using the HoloLens was rather straightforward; we had access to one at our office and it is, in our opinion, far ahead of other AR HMD’s on the market. Unity was chosen as our development environment because it has well integrated MR solutions and frameworks such as Microsoft’s MRTK and Google’s ARCore-plugin. The paper “Design for Collaboration in Mixed Reality” cites it as the most “cost-effective, flexible and sustainable solution” for developing MR applications [35]. This fact, together with a strong developer community, made it the best choice compared to the other options we had considered. As we came to this decision, we also realized that we had to limit our scope. Due to a lack of both time and resources, we were forced to leave the VR integration aside for the time being, and to solely focus on AR.

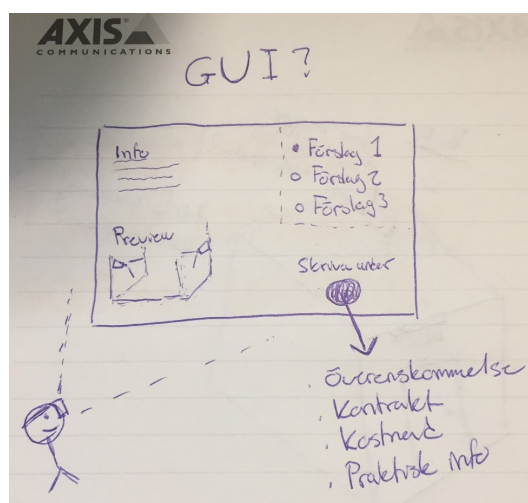


Figure 3.2: A quick sketch of a holographic contract.

In order to speed up development, the project was split into two parts - one focusing

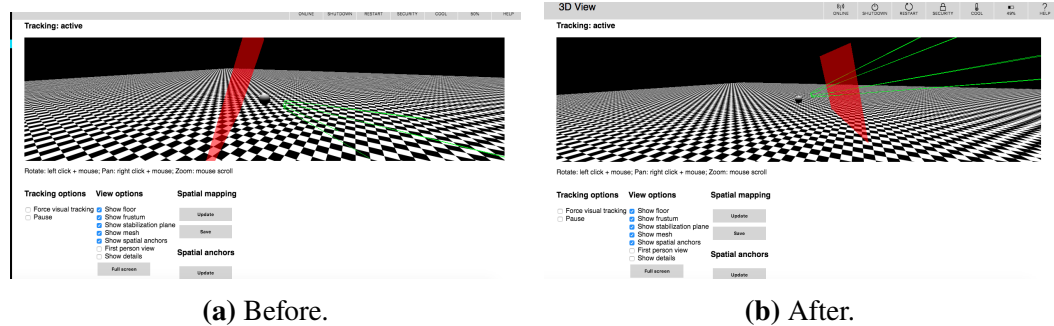


Figure 3.3: The stabilization plane before and after the depth buffer was shared.

on building and processing the model of the room, and the other one experimenting with different interaction methods. To enable this split, we used Git to collaborate, share and track our code, together with Git LFS to avoid storing 3D models and other large files on our Git repository. When facing application-breaking changes, we utilized branches in order to continue working without interruptions. In the beginning we were worried that Git and Unity was not a good match due to large, binary files, scene settings and so on. But fortunately, this setup worked very well and did not cause us any trouble.

Developing AR applications to the HoloLens is not always well optimized, and at times it can be somewhat tedious. Deploying new builds to the HoloLens takes time and it is not uncommon having to deploy continuously during a short period of time to get small details right. The Unity editor, together with the MRTK, allows for simulation of HoloLens applications directly in Unity. This definitely eases development, but sometimes there is a mismatch between the simulation behaviour and the behaviour on the actual HoloLens which can cause problems if realized too late. Additionally, some things are not optimal to simulate in a game editor, such as spatial mapping and tag along functionality. One of these issues showed up when we tried to record a quick video through the HoloLens. As soon as we started to record, the holograms started moving around the space, no longer anchored in the world. After some research, we found that the latest versions of Unity required you to explicitly allow access to the so called "depth buffer", which feeds data to the stabilization plane. The stabilization plane is crucial in keeping the holograms stable. When debugging, we found it to be behind the user, instead of in front like it is supposed to be. This is demonstrated in Figure 3.3

3.1.3 Finalizing the application and User testing

When both parts of the application had started to take form, they had to be merged. It turned out to be a smoother process than expected, and days after we started merging, we had everything up and running again. From here on out, some smaller features were added, but in large it stayed the same.

When development were completed, we moved on to general user testing. Two slightly different versions of the application were created to compare different UI interaction methods. The differences will be further explained in section 4.2.1. For more information on how the testing was done, see chapter 5.

3.2 Feedback and User-Centered Design

User-centered design (UCD) is a framework that is often used when developing systems or applications that eventually will end up in the hands of an end user [32]. While this thesis' work process has not fully enforced UCD, due to its research-focused nature and the frequent change of directions, we still acknowledge the importance of continuous user feedback. Therefore, we have taken every opportunity to demonstrate and discuss our prototype during the course of the project. Additionally, the second half of the project was almost exclusively focused on user testing and usability evaluation, to gather data and feedback for further development.

We had meetings biweekly with our supervisors where we discussed what had happened since the last meeting and what we planned to do next. A lot of the time were dedicated to testing and evaluating new features, finding bugs and letting our supervisors request improvements or additions to the prototype. A typical meeting can be seen in Figure 3.4.



Figure 3.4: A meeting with our supervisors.

Since all three of our supervisors are part of the EASE (Embedded Applications Software Engineering) project [19], we took the opportunity to gather some early feedback on one of their meetings. EASE is a collaboration between academia and industry where “the objective is to ensure that industrial partners have a competitive advantage with respect to competency and innovation of novel solutions and effective engineering of embedded software applications with physical and logical mobility”[19]. We got the opportunity to demonstrate our prototype and afterwards we received feedback from a UX expert at Sony and an associate professor at Lund University with expertise in AR and VR.



Figure 3.5: Axis Global Sales Engineers Conference.

We have also gotten the opportunity to present our work and gather feedback at a number of internal Axis-events. For example, we presented our work to another Axis department at their monthly “innovation hour” meeting. They were very interested in how one could integrate camera feeds in AR. The discussions we had at the meeting drove us to implement virtual camera feeds into our prototype, which turned out to be very handy. In addition to this, we also hosted a “halfway presentation” for the CTS department, and took part in Axis’ Global Sales Engineers (GSE) conference, where we got the chance to talk to the people actually selling the camera systems, which is shown in Figure 3.5.

Chapter 4

Application development

This chapter will describe the application itself, and its many moving parts. It will begin by giving an overview of the application and how it was designed, and then follow up with descriptions of its different parts.

4.1 Ideas and requirements

When development of the application began, there were several requirements to be fulfilled. Some of these requirements differed considerably, and could not coexist very well. For example, a user should be able to manipulate and examine different camera models freely, i.e move, rotate and scale, in the 3D space. Additionally, the application should place cameras in optimal positions, based on the room which the user currently interacted with. This would to some extent restrict a user from moving a camera freely, which contradicted the first requirement. As a result of several contradictions like these, the development of the application was divided into four main parts, which more or less were developed separately. The different parts are described in detail in this section, but they can briefly be summarized as:

1. Interacting with camera models in 3D space.
2. Creating a model of a real room and suggest camera placements based on it.
3. Simulate a security system and its functionality.
4. Visualize what a camera sees.

At the end of the development cycle, three of these parts (2-4) were integrated into a single application which followed a step-by-step scenario.

4.1.1 Interacting with different camera models

An important first step is to show the virtual cameras in the real world, and introducing users to how they can be interacted with. Three 3D models of different types of Axis cameras are visible to the user. They can be manipulated and examined freely in the 3D space. Above each camera there are two radio buttons, where either rotation or movement can be selected depending on what kind of interaction the user wants to perform. An Axis dome camera with the associated radio buttons is shown in Figure 4.1.



Figure 4.1: Preview of a camera model in Unity.

The cameras also have a world anchor attached. A world anchor is a tool to map a virtual object to a specific position in the real world. This allows the cameras to remember their position in between application sessions, which means they will always remain at the spot where the user last left them, even if the application is restarted.

In this part of the application the user also has access to a UI element that is called “the contract”. This is intended to give some information to the user regarding the camera models, price suggestions and so on. The user can also switch between different kinds of camera solutions from the contract.

When support for two handed manipulation got introduced in Microsoft MRTK, a separate demonstration project was created to allow users to interact with camera models with the two handed interaction. This project was built very similar to what is described above, but since the projects used two different versions of the MRTK, there were too many compatibility issues to merge them into the same application, which means that the current version of the main application does not support two handed manipulation. This small demo was used when discussing how to interact with holograms in the future.

4.1.2 Mapping the room and building a World in Miniature

To bridge the gap between the virtual and real objects, the application needs to gather information about the environment. This part of the application is focused on building a model of the room a user interacts with, and to shrink it down to a World in Miniature (WiM). A WiM is a minimized, virtual copy of an environment which could be interacted

with, or be used as an overview [42]. The idea is that both the actual size model and the WiM could be used in the application, depending on what the user wants to do and the situation. Additionally, a big reason for implementing this functionality is to, in the future, be able to export the models in order to work with them in different environments, like a desktop computer, or to share them with someone else.

To map a room, the user walks around the space and scans it by looking around, using the built-in spatial mapping capabilities of the HoloLens. When an area is successfully scanned, it will be marked with a green mesh overlay, to make it easy for the user to understand which parts of the room that are mapped. When the user is satisfied, the mapping stage can be ended and the application will finish up the model, filling any holes and smoothing inconsistencies in the mesh. At this point, the WiM will be generated from the produced model. The application will also give suggestions for camera placements, by generating a couple of green boxes in the room. The intelligence of the suggestions is very limited in the current version. It places the indicators on random points in the room, as long as they are on a wall and at least 40 centimeters apart from any other indicator. This is only meant to demonstrate future capabilities.

At this point the user can create cameras and place them in the room. The application will give feedback regarding the placement to the user by simple color coding. The camera will turn green if being placed on the suggested indicators, it will turn yellow if being placed on a suitable area but not on an indicator and it will turn red and prohibit placement if aimed at an unsuitable area, such as a floor or a table. The different scenarios are shown in Figure 4.2. The placed cameras will be visible both in the real scale room and in the WiM. Figure 4.3 show how the WiM looks with a couple of cameras in place.

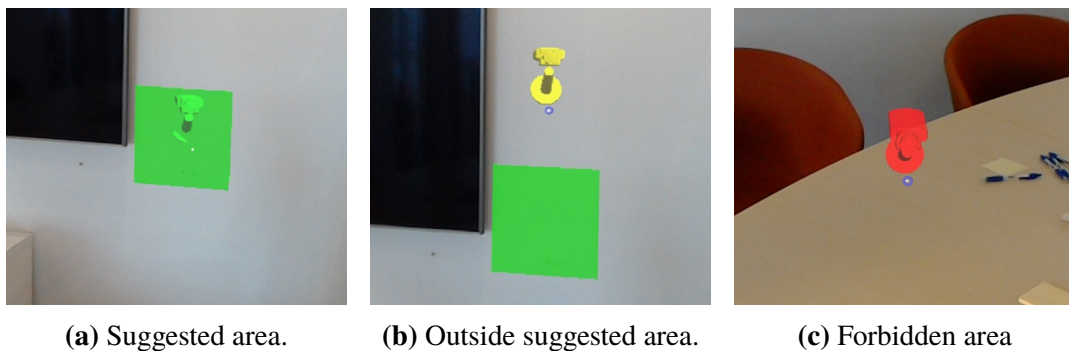


Figure 4.2: The color of the camera changes depending on what surface and position the user tries to place it on.

4.1.3 Simulate a security system

An important requirement for the application is to preview or simulate a security system, in order to help potential customers get a better understanding of it. In its current version, the application supports a simple use case of a person walking in the room, and the system reacts when this person passes a cross detection line. Currently, this person is represented by an animated avatar. In the user testing, we choose to call the avatar Bob, he can be seen in Figure 4.4. Bob can be freely positioned across the floor by the user. To get Bob to move, the user constructs a path by placing out a number of waypoints in the room. When

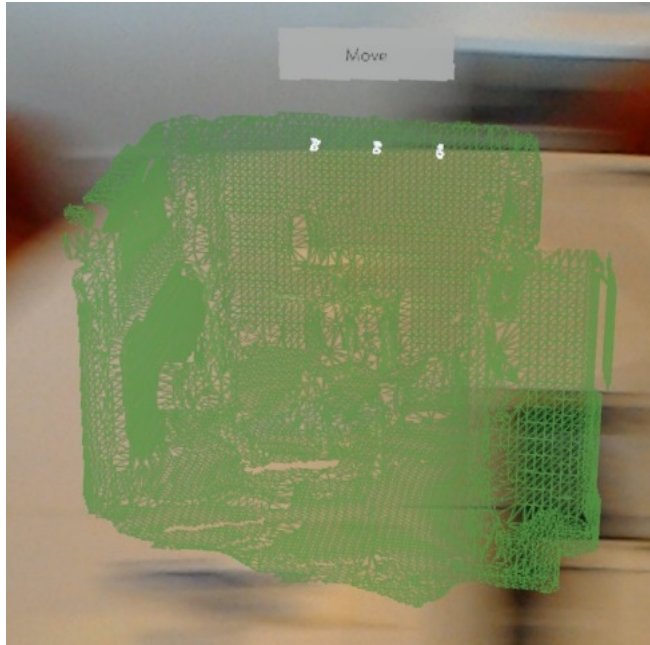


Figure 4.3: The WiM model with cameras.

Bob is activated, he will move along the placed waypoints in the order as the user placed them. It is also possible to place any number of cross detection lines in the room, which will trigger a warning if Bob happens to cross any of them.

4.1.4 Visualize what a camera can see

Lastly, the application aims to help a user understand what a camera feed can capture and how to adjust the camera based on this information. In the top right corner of the user's vision, a small virtual display is placed to show the feed of a camera. Figure 4.4 shows how this looks like while wearing the HoloLens. Unlike the other holograms, this screen is stationary on the HoloLens, and cannot be moved. It can however be enlarged or hidden.

To change which feed is displayed, the user can click on the corresponding camera, or choose it from a menu, seen in Figure 4.5. The camera with the active feed will always be colored blue, while the other cameras are white.

The user can also make adjustments to a camera and see the changes directly through the feed. By choosing "Modify Camera Position" in the menu, the feed display is enlarged and centered, and adjustment UI elements appear, see Figure 4.6. In this mode it is possible to pan, tilt, zoom and move the camera along the wall.

4.2 Interacting with the application

In addition to the different gestures and methods described in 2.5.2, it is also possible to interact with the application by UI elements and voice commands.

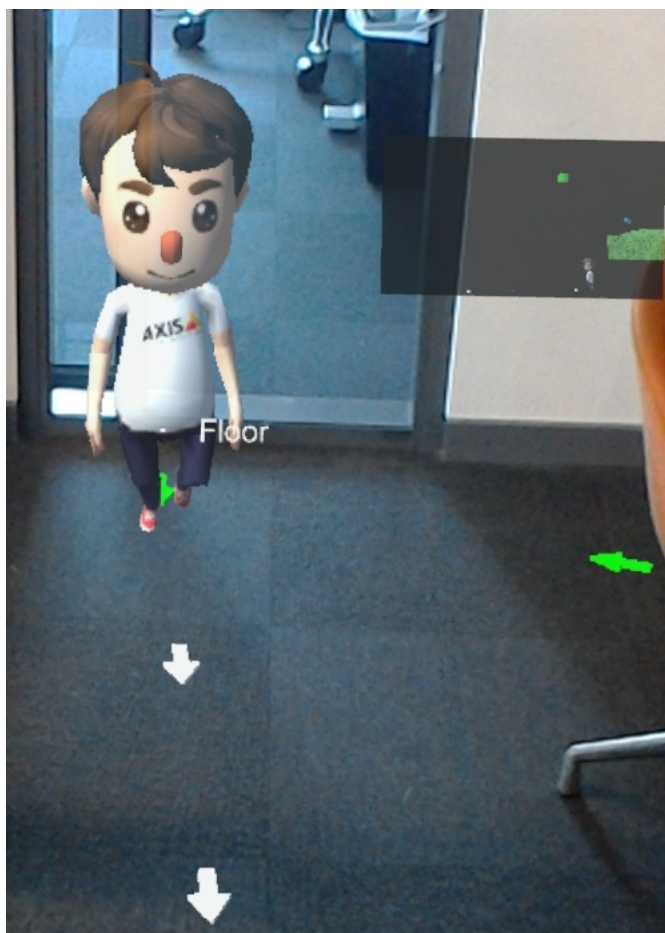


Figure 4.4: The point of view of a user wearing the HoloLens, while Bob is walking along a path. In the top right, the camera feed is shown.

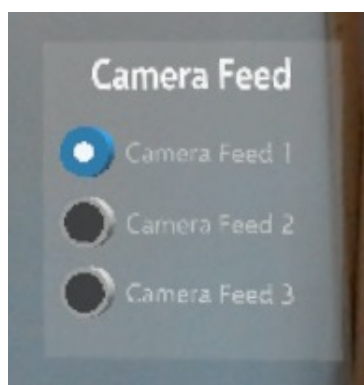


Figure 4.5: Menu for selecting camera feed.

4.2.1 UI elements

To make functions and tools available in the application easy to find, a UI with a couple of submenus was created. The main menu, with its two primary submenus, can be found in Figure 4.7. The functionality for handling camera events, and for handling events tied to



Figure 4.6: The UI used to modify camera settings.

Bob are split up between these submenus. In the mini-display menu it is possible to switch camera feeds, toggle and enlarge the mini-display and modify camera and feed settings. In the avatar menu it is possible to create waypoints and lines to enable cross line detection, interact with Bob and to start a simulation of the system. There are two versions of how the UI interaction can work. In one version the menu will always remain in a user's vision. If the user slightly move his gaze, the menu will stay in place, but as soon as it reaches the peripheral area of the vision it will tag along with the user. In the second version, it is possible to move and place the menu freely by dragging the green square that is tied to the menu. The green square can also be clicked on in both versions, to minimize the UI. The square then turns yellow and a second click will make the menus appear again.



Figure 4.7: The menu with submenus extended.

4.2.2 Voice commands

Some functionality in the application can be triggered by voice commands. Below the most relevant voice commands are listed:

- New point: Create a new waypoint and puts it in tap to place mode.
- New line: Create a new line and puts it in tap to place mode.
- Start: Activate the simulation of the system; Bob starts to move along his path.
- Stop: If Bob is moving, he will stop and wait for the simulation to be resumed.
- Feed <1,2,3 >: Change the active camera and the active feed to the desired camera.
- Modify camera: Open the GUI to modify camera settings.

There are additional voice commands available in the application, but many of them are used mainly for debugging or testing purposes and are not relevant for an end user.

Chapter 5

Final evaluation

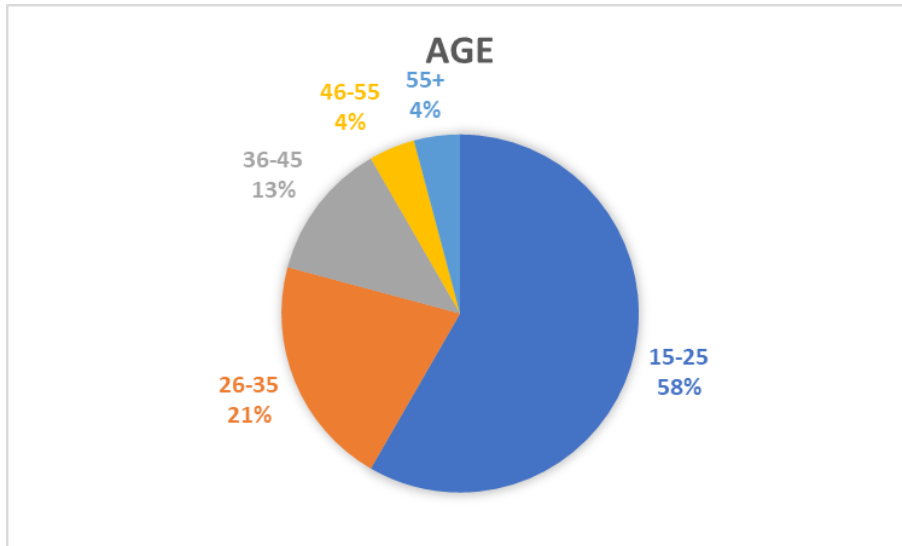
This chapter will describe the user study; the purpose and goal, information about the participants, the test setup, and how the questionnaires were constructed. Furthermore, the user test execution will be described and finally, the results from the questionnaires and the interviews will be presented.

5.1 Purpose and goal

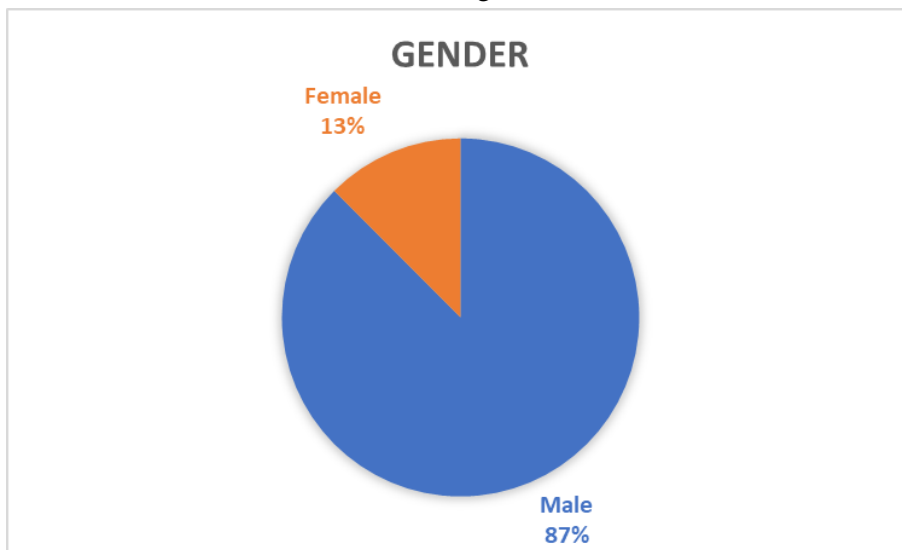
The purpose of the user study was to evaluate the general usability and workload of the application and to gather feedback for further development. Additionally, we wanted to compare two different methods of controlling important holographic UI elements. Lastly, we hoped that carrying out user tests could spark new ideas for how to visualize and configure a security system with AR.

5.2 Participants

In total, 24 participants and two pilot testers took part in our user study. The participants' division of age and gender as well as their experience with AR can be seen in Figure 5.1 and 5.2 respectively. A participant was considered to have prior experience with AR if the participant had used an application with some kind of AR implementation. The kind of device used or the amount of experience was not considered for this categorization. In other words, even a one time use of a smartphone AR application could be considered as prior experience. The same reasoning goes for the experience with HoloLens. The amount of time spent with the HoloLens and the complexity of the application were not considered for the categorization. Some participants did not have experience with HoloLens but with other AR headsets.

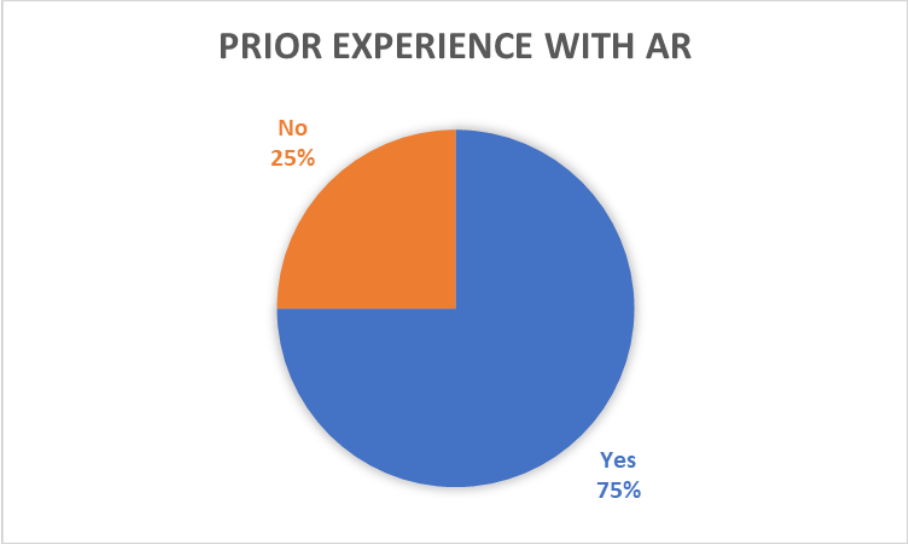


(a) Age

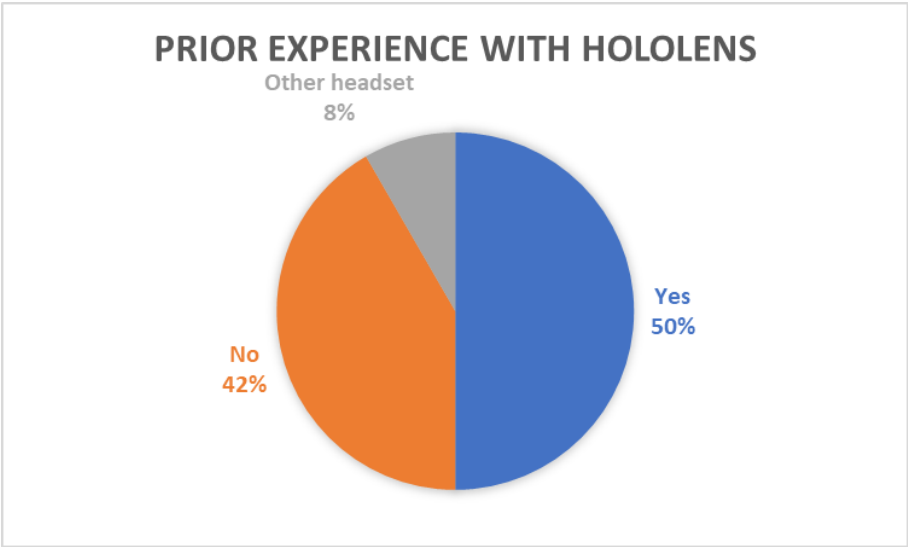


(b) Gender

Figure 5.1: Demographic information about the participants.



(a) The participants experience with AR.



(b) The participants experience with AR headsets.

Figure 5.2: Information about the participants prior AR experience.

5.3 Test setup

Due to some restrictions with room availability, we carried out our user tests at two different locations based on the test participant. If the tester was an Axis employee, the test session took place in a conference room at Axis. If not, the test session took place at Ingvar Kamprad Designcentrum (IKDC), in a Cave Automatic Virtual Environment (CAVE) room. None of the CAVE's features were active during the tests. The two rooms were chosen because they had similar structure and size. The CAVE room can be seen in Figure 5.3 and 5.4. The sessions were recorded with the HoloLens feed capture and with an external video camera. Unfortunately, the HoloLens recording was very unstable at times due to network instability, which forced us to skip that part for several of our test sessions.

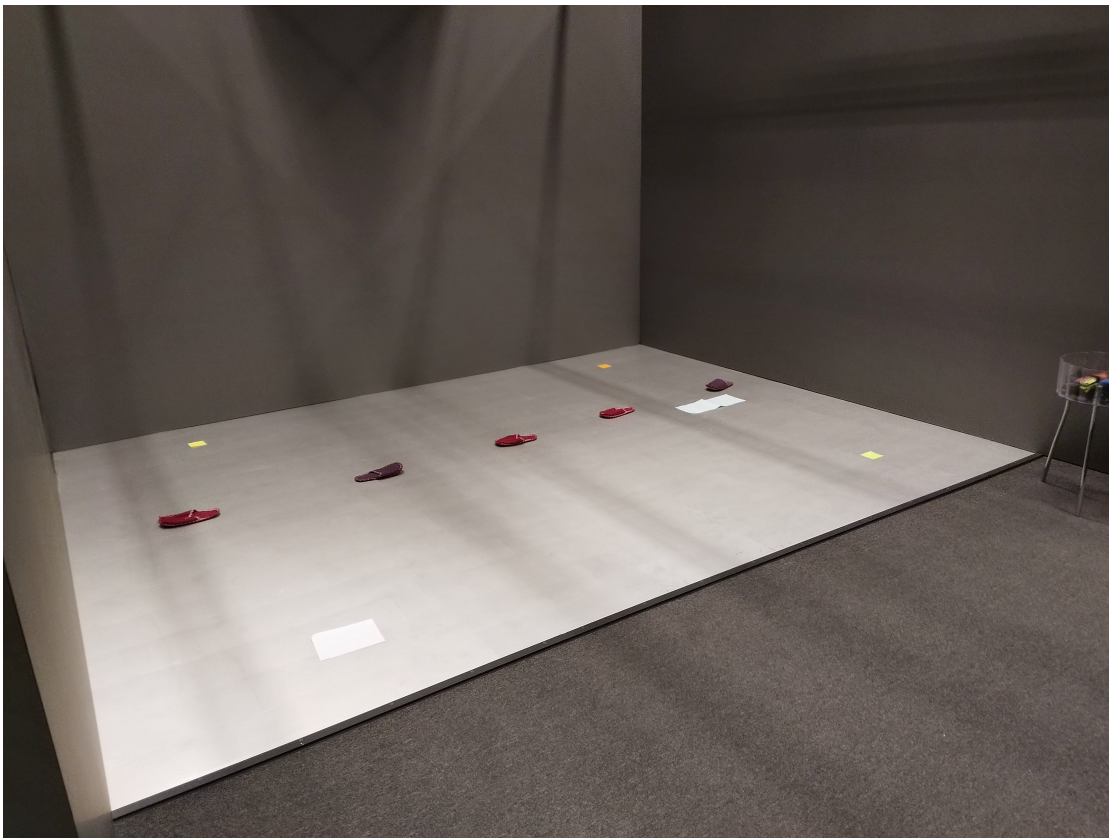


Figure 5.3: The test setup in the CAVE at IKDC.

A couple of markers were placed out in the room to indicate where certain holograms should be positioned by the user. There was one marker for Bob's position, one marker for the cross line detection and three markers for the waypoints that would make out Bob's path. This was done to simulate fixed requirements for the system, which the tester should implement with the help of the application. Each test participant was given a paper with step-by-step instructions to follow during the experiment. The entire instruction text can be found in Appendix D. We had two slightly different versions of the application, to evaluate which version was the most efficient. The difference was how the user handled the UI elements in the application. The first version let the user move the UI elements and in the second version the UI elements always followed and stayed in the user's FOV.

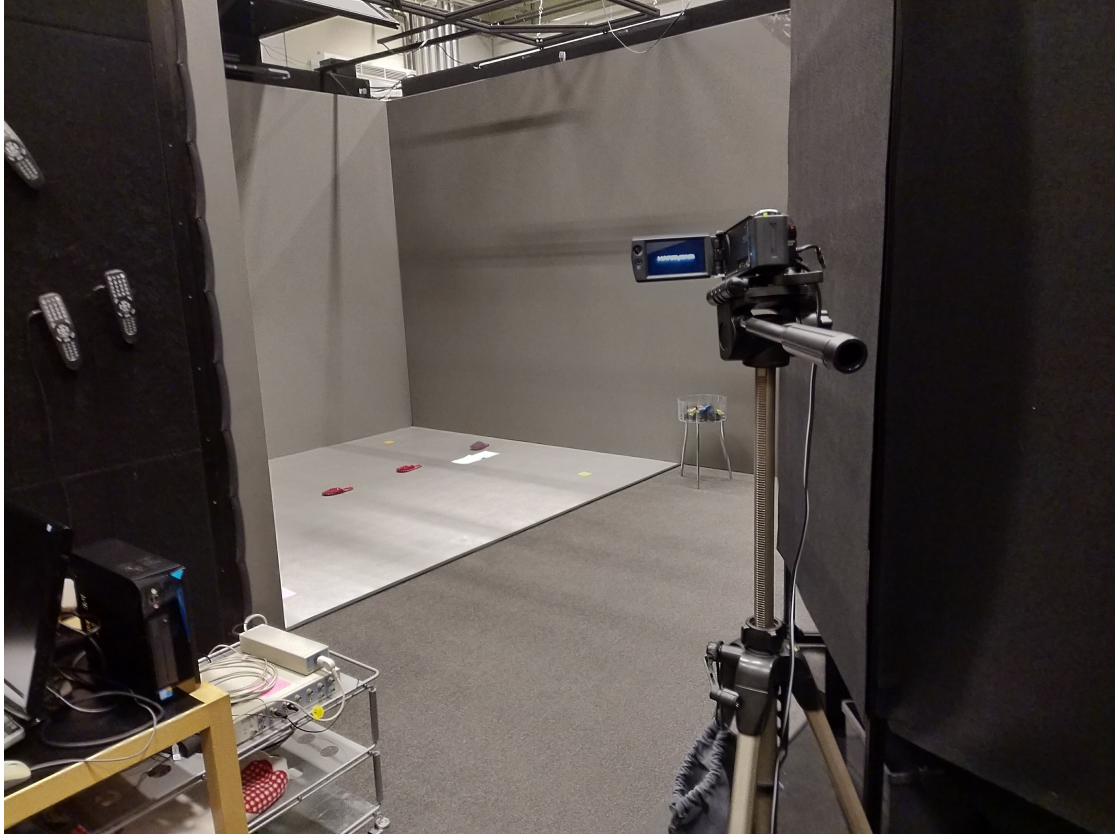


Figure 5.4: A stationary camera was used to capture the tests.

5.4 Questionnaires

During the test session, the participants were asked to fill out three different forms. The first one, which was filled out before the experiment began, was a form to gather personal data and information about the overall AR experience of each participant. When the test was concluded, they were asked to fill out two evaluation forms; a slightly modified versions of SUS and the NASA RTLX (Raw TLX). We have found that the second part of NASA TLX is time consuming and that participants tend to get unfocused after a couple of weighting choices, and it has been shown that skipping the weighting part altogether has the same sensitivity as the original version [17, 21]. This is why we are using the RTLX method, in which only the workload ratings are used to calculate an average workload score. To gather and export this data we used an online NASA TLX tool [40, 39].

Four of the ten original SUS statements (1, 4, 9 and 10) [8] was modified so that it would fit our context better. We have seen discussions regarding doing so in other cases as well [37], and we think these modifications are necessary and believe that they will not affect the outcome to a large degree. Two of the modified statements (1 and 4) are only slightly rephrased so that the participant are more likely to rate based on our application, rather than the HoloLens. The other two statements (9 and 10) are more clear modifications since it would be hard for the participants to separate the application experience with the HoloLens experience when answering those. However, they are modified in such a way that they are still covering the same “field” of usability. In our pilot tests we kept the

original questions, and even though the sample size was very small, we could see clear indications that the ratings were skewed towards that the participants rated the HoloLens experience rather than the application in the original statements. In our actual tests we chose to remove the original statements and replaced them with the modified ones. The modified SUS questionnaire can be found in Appendix B.

5.5 User test execution

The application was to a large extent developed with a use case scenario in mind, and therefore it seemed reasonable to build our user tests according to that use case scenario. Below the different steps will be briefly explained, and in Figure 5.5 the different stages of the test session can be seen. The full test plan can be found in Appendix E.

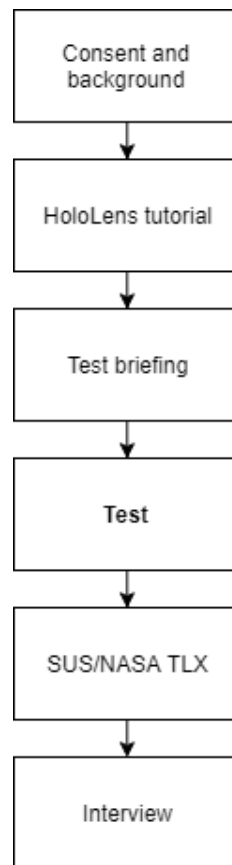


Figure 5.5: The flow of a test session.

HoloLens tutorial

The group of test participants included a large amount of people that had never used a HoloLens and were unfamiliar with AR. To teach every participant the basic gestures and functionality of the HoloLens, a simple tutorial application was created, which every test user got to try out before the actual test session began. The tutorial had the additional benefit that different interaction implementations found in the main application could be

introduced beforehand. In the tutorial, the participants could interact with a sphere and a cube. They were asked to move around the room to get an understanding of how the holograms were anchored in the world. The sphere introduced the tap to place interaction, which is used to place most holograms in the main application. The cube introduced the tap to select and hold to drag interaction which is used to choose which camera should be active and to manipulate a camera's and the main menu's position in the actual test.

Test briefing

After the participants felt comfortable with the basic interaction methods, they were briefed on what they were going to do, some background info, and general information about the test. The manuscript for the briefing can be found at in Appendix G.

Test scenario

The users should first scan the room, and a miniature model of the room is then created and the camera positioning is prepared. When the scan is done, the application places six visual indicators on the walls to mark the suggested camera positions. The users can create three cameras, and should place each of them on any of the indicators. Finally, the users should set up the system with the help of AR to get a feel for how it would work in reality, and simulate a sequence of events to verify that it works. The scenario can be read in its entirety in Appendix D, and a simplified version with all the tasks can be found in Figure 5.6.

SUS and NASA TLX

Next up, the users evaluated the application using the SUS and NASA TLX questionnaire on a laptop located in the respective testing environments. The SUS questionnaire can be found in Appendix B.

Interview

After each test session, a semi-structured interview was conducted. The participants were asked to describe their feelings towards the application and its usability. Additionally, a couple of predefined questions regarding UI handling and hologram interactions were asked to all participants. The full interview manuscript can be found in Appendix C, and some important takeaways will be presented in section 5.6.2.

5.6 Test results

All the participants completed the test, with the average time 15:23. The rest of the results from the questionnaires and the interviews will be presented in the following sections.

Task	Sub tasks
Initialize the model and create the WiM	<ol style="list-style-type: none"> 1. Scan the room (walls, ceiling, floor and objects). 2. Finalize the model mesh. 3. Create the WiM.
Place cameras	<ol style="list-style-type: none"> 1. Grab a camera 2. Place the active camera on a wall, preferably on a suggested spot (visualized with a green overlay). 3. Spawn a new camera. 4. Repeat step 1-3 until three cameras are placed in the room.
Interact with the cameras	<ol style="list-style-type: none"> 1. Select a camera that is not active. 2. Adjust the camera position by dragging it so that the avatar (Bob) is visible in the feed. 3. Zoom out with the camera, using the adjustment menu.
Set up a path for Bob to follow	<ol style="list-style-type: none"> 1. Place Bob at his starting position (user's choice). 2. Place the existing waypoint as Bob's first target to reach. 3. Spawn an additional waypoint (by clicking the UI button, or by voice command), and place it as the next target in the path. 4. Repeat step 3 until at least four waypoints has been placed.
Place cross line detection along the path	<ol style="list-style-type: none"> 1. Place at least one cross line detection (by clicking the UI button, or by voice command) somewhere on the path created with the waypoints.
Start the simulation	<ol style="list-style-type: none"> 1. Start the simulation. 2. See if the system acts according to expectations.

Figure 5.6: The different tasks that were performed by the participants

5.6.1 SUS and NASA TLX

According to Bangor [6] our application would be rated as between Good and Excellent with an average SUS score of 78.13. Version 1 had an average score of 77.08, while Version 2 had an average score of 79.17, see Figure 5.7. As further reference, Jeff Sauro analyzed data from 500 studies where SUS was used [38]. He found that the average score across these were 68, and concluded that score to be a general reference to rate systems above or below average.

According to Grier [15] our application would be rated in the 20% percentile with an average workload score of 28.05. Version 1 had an average Workload Score of 31.31, while Version 2 had an average score of 24.79, see Figure 5.8. To put the workload score

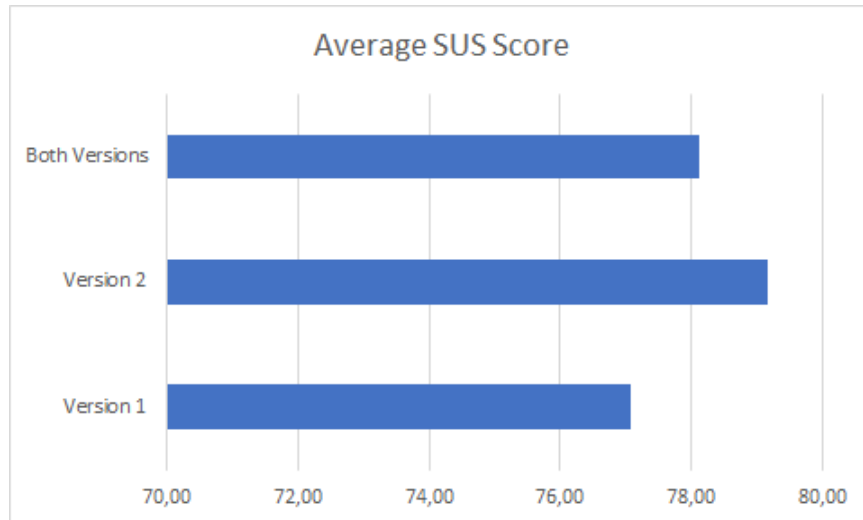


Figure 5.7: The average SUS Score for both versions, versus the SUS Score of the individual versions (higher is better).

in further context, the mean of all results gathered by RTLX in [15] was 45.29. There were two data points for doing "no task" which got workload scores of 12.0 and 14.8.

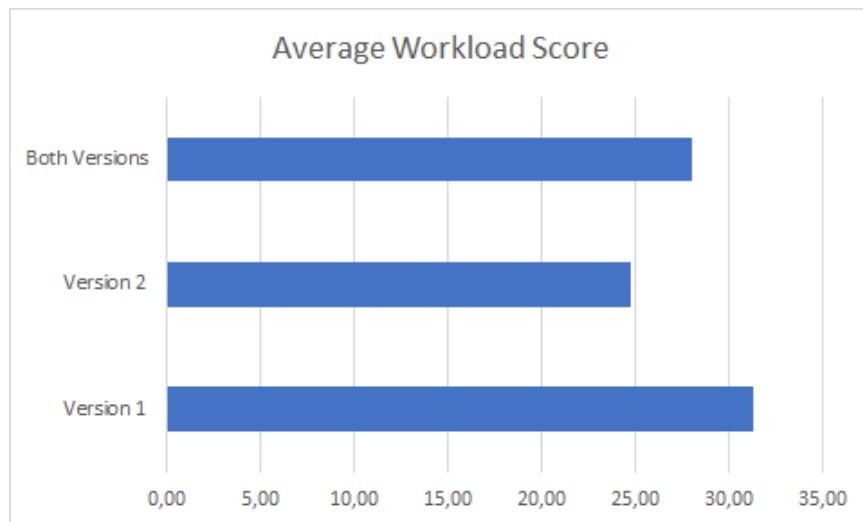


Figure 5.8: The average workload score for both versions, versus the workload score of the individual versions (lower is better).

Figure 5.9 show the workload score broken down into the individual categories. It also shows the minimum och maximum value for each category. Figure 5.10 demonstrates the median workload score for the two different versions, also broken down into the individual categories.

Figure 5.11 compares the average time of experienced and inexperienced users. Users were classified as experienced if they answered that they had studied or worked with AR professionally in the initial questionnaire.

Figure 5.12 display how users rated the relevance of the application in the SUS questionnaire. Every participant rated the relevance as high. On the 5 point scale, 16.7% rated

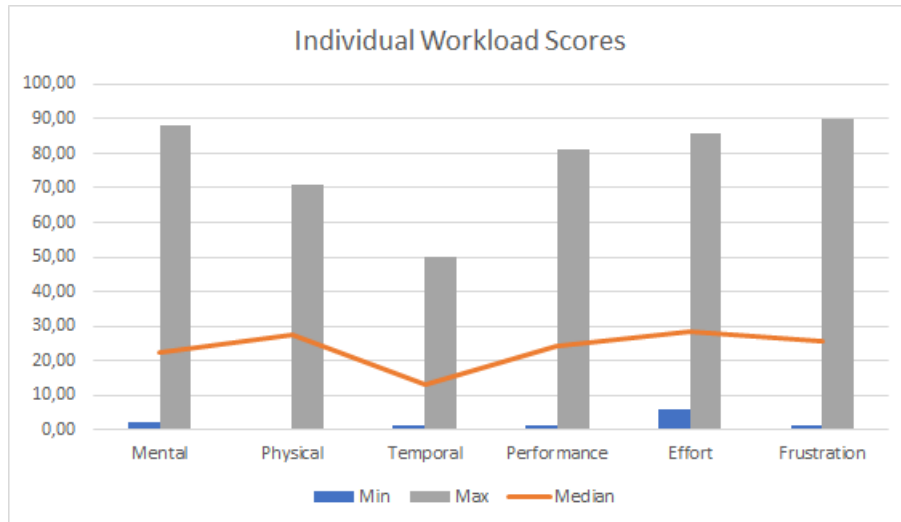


Figure 5.9: The minimum, maximum and median workload score of all the individual categories (lower is better).

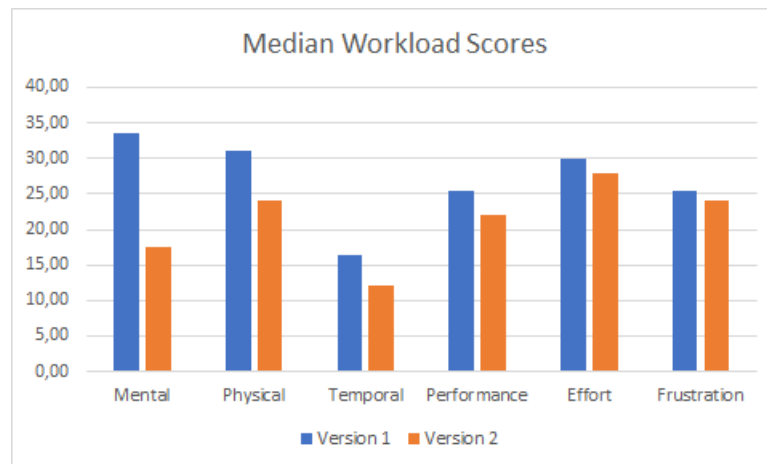


Figure 5.10: The median workload score of all the individual categories, Version 1 versus Version 2 (lower is better).

4 and the rest, 83.3%, rated a 5.

5.6.2 Interviews

This section will present the main takeaways from the post test interviews.

Interaction and UI

Almost all of the participants described the UI as "simple, easy to understand and relevant to the context". Some thought that it took some time to get used to the menus, but chalked that down to it being the first time they wore the headset.

When asked about what they thought about the UI version they got to use, a majority answered that in theory they would prefer the version where they got to place the menu by

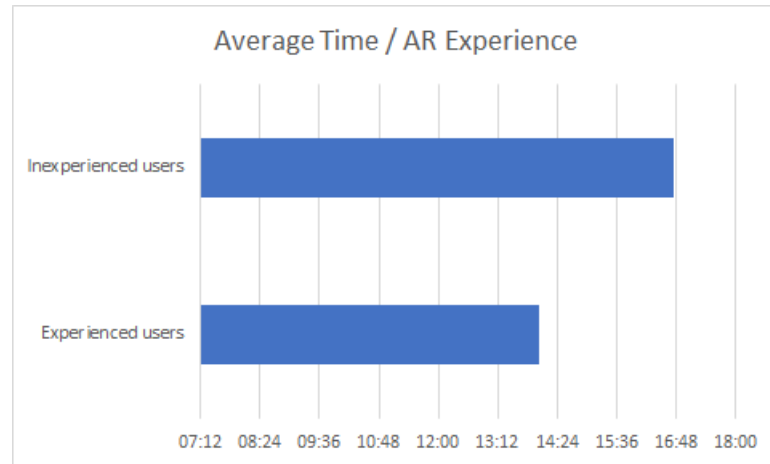


Figure 5.11: The average time for inexperienced and experienced users.

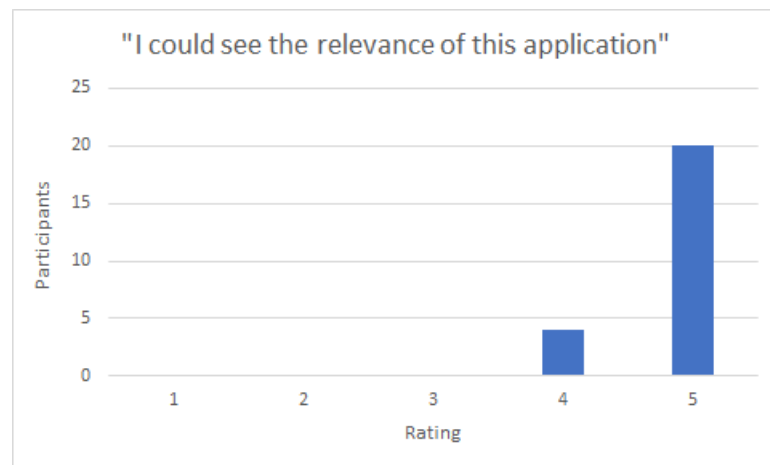


Figure 5.12: The test participants' rating of the application's relevance.

themselves.

Another thing that came up several times throughout the interviews were the lack of indicators. Many found it confusing and frustrating when things appeared without their knowledge, and had a hard time keeping track of objects. Some people referenced video games, where indicators are often used to show the location of important objects off screen.

Camera placement

Just like the UI, most of the participants thought that the camera placement was simple, but it took some time to get used to initially. Many of the users saw the recommended placements for the cameras, but not everyone used them. The reasons for this varied, some people were so focused on building an optimal system that they themselves wanted to decide where the cameras were placed, and some people simply did not understand that they were meant to place them there.

Some participants commented on the fact that the first camera was spawned in imme-

diately, together with the buttons text being "Create Camera 2", confused them. The big button demanded a lot of attention, and therefore they missed the camera itself.

Additionally, some wondered why this step even existed in the first place. Why not place the cameras in their recommended position first, and then let the user move them around?

Scanning the room

Most of the participants found this step to be easy and straightforward. They thought they were given good feedback on what parts of the room they had scanned, and when they could move on to the next part.

However, many had trouble deciding when the process was done, citing the small FOV as one of the reasons. A requested feature was some kind of indicator, like a progress bar, to determine when the scanning were nearing completion.

The WiM

Although the WiM was mostly left out of the scenario itself, participants could still interact with it. Most people noticed it, and some of them also noticed the fact that the cameras were placed in the WiM as well as in the real environment.

The ones that noticed the cameras liked it, saying that it made things easier, and that it gave them some much needed overview.

General thoughts

Many praised the idea behind the application, although some found it somewhat lacking in execution. A lot of the problems the participants had could be traced back to the limitations set by today's technology, and in particular the HoloLens. The limited FOV and cumbersome gestures came up several times.

Chapter 6

Discussion

In this chapter we will analyze the test results. Additionally, the concept of using AR to implement security systems will be discussed. Furthermore, the different interaction methods that were implemented will be motivated and discussed.

6.1 User study

The results from the user study shows that the version with a tag along UI (Version 2) performed better than the UI with manual handling (Version 1). This was true for both the SUS and NASA TLX questionnaire, though the difference was smaller in the SUS evaluation. This could mean that manual handling of menus and UIs affects the workload of a system to a larger degree than it affects the usability. The result was somewhat surprising, since a majority of participants stated that they would prefer Version 1, when asked in the post test interviews. A likely explanation could be that since each participant only tested one version, they misjudged how important it is to maintain control. As control is generally seen as something positive, users might overrate it and feel reluctant to hand it over to the application, before they understand how it works and what it does.

The results also show how users experience different kinds of issues, or fail to notice certain problems. One of the main complaints by participants using Version 2, was that the UI sometimes blocked other objects, forcing them to slightly adjust their head to move it out of the way. It is something that is immediately inconvenient, and requires the user to take action in order to solve it. Version 1 on the other hand, received less complaints, most of them regarding the fact that they had to remember where they had placed the UI, and had to return to that spot time and time again. This might not immediately be recognized as a problem by the user, even though it requires more physical movement than Version 2. This could lead to the participants rating the workload higher for Version 1, without being able to identify what caused it. This theory is also supported by the results of the individual subscales of NASA TLX; the workload scores for both mental and physical demand are

notably higher on Version 1 than on Version 2.

As presented in section 5.6, the average SUS score was 78.13, and the average workload score was 28.05. Both scores are regarded as above average, and indicates that our application has good usability and low workload. While developing the application, much thought was put into the usability aspect, since we expected that simply using an AR application on a HoloLens would challenge our users. If the application was not easy enough to use, it ran the risk of being rated way below average on both scales.

One reason for the high ratings could be that users were dazzled by the technology. Even though our data states that 75% of the participants had prior experience with AR, using a HoloLens is a completely different experience than using AR on a phone. Of the 50% of users that had prior experience with the HoloLens, many of them had only tried it on once or twice for a few minutes, so their experience were very limited. Based on this we believe that some users did not know of the HoloLens' potential, did not have much to compare with, and therefore were impressed by a working prototype. Additionally, the scenario itself was not very strict, allowing the users to complete the tasks without much oversight from us. This meant that every user completed the test successfully, and the main indication of how well it went, both for us and the user, was the time spent. This could cause the users to overestimate theirs and the application's efficiency. In hindsight, implementing some sort of rating system to grade a test session is something that could have generated some interesting data.

Even though we would have liked additional parameters to measure how well a user performed, the time it took to complete the tasks still seem to be a rather good indicator. We base this on the test results seen in Figure 5.11, where the most experienced users had the fastest completion time. That experienced users would perform better and have an easier time was what we expected, and these results seem to confirm that theory.

6.2 Using AR to implement security systems

We interpreted the result regarding the application's relevance shown in Figure 5.12 as a clear indication of that AR has great potential in visualizing and implementing security systems. We consider our application an early prototype, which makes it even more exiting to receive this kind of feedback. One of the strongest arguments for using AR in this context is to be able to preview solutions without having to rely on physical cameras. Mistakes or obstacles are easier to realize early on in the planning process, before actual equipment is ordered or an installer is sent to the site. The ability to share solutions, or even show them in real time between the different parties is also a huge selling point. This could mean less traveling to physical locations for everyone involved, which both saves time and money. It also allows for a customer to get a full scale simulation of certain scenarios, which can enhance the understanding of the system, and increase the trust in the system's functionality. If customers can confirm the functionality of a system themselves, deals could potentially be faster and more efficient for all parties involved.

Many of the issues we detected during our user tests had to do with the HoloLens' limitations. For instance, several test participants had problems with; the small FOV, loss of tracking, difficulty to put the headset on and disturbing light sources or reflective areas.

These problems are often the main concerns people have when discussing AR appliances in our context. The good news is that these issues can most likely be solved or mitigated as the technology and the hardware matures.

6.3 Interaction methods

One of our goals with the thesis was to experiment with different AR interaction methods, presented in section 2.5.2, and they are all implemented in the final version of the application. In this section we will discuss and motivate why and how we chose to use these methods.

6.3.1 Hold to drag

This method was intuitive and easy to understand for the participants, most likely because they are used to the same interaction metaphor with computer mice. It can be hard to use this method in combination with other object manipulations, such as rotation, unless some kind of toggle functionality is in place. While rotating is possible by adjusting the head while holding an object, it often feels unintuitive and uncomfortable. This also makes users rotate objects by mistake, so we chose to completely forgo this behaviour when implementing the hold to drag method.

We use hold to drag in our application when a user should be able to move objects freely in the room. Our reasoning for this is that we believe most people will carry the computer mouse drag and drop metaphor with them, even when using other kinds of software or technology. Additionally, we have a constant camera feed of the camera that is being moved, and the hold to drag method provides smooth movement. This means that the feed image also updates smoothly as the camera is moved around, which provides the user with valuable on-the-fly feedback on how the camera movement affects what the camera can see.

6.3.2 Tap to place

Using tap to place to move objects might not be as intuitive as the hold to drag method, but when a user gets a hang of it, it is much more efficient. We use it when the object needs to follow certain rules when placed. It is especially useful when moving objects a large distance, since it requires less movement than hold to drag, and there is no risk of losing the object by accidentally moving it outside of the gesture frame. Additionally, spatial mapping can be used in conjunction with the tap to place gesture, allowing objects to be snapped to mapped surfaces. This is a great functionality in our case, since we often want to place objects on specific parts of the room.

6.3.3 Toggle method

This method is used in combination with other gestures to toggle between different interactions such as moving, rotating and scaling. It can sometimes feel a bit slow, having to

take an action every time you slightly want to rotate or scale an object. In our application, this method is also used to switch between zoom, tilt, pan or move, when a user modifies a camera's settings with the adjustment UI. We noticed that it is important to have big UI buttons that gives clear feedback when pressed. Missing a toggle click can cause additional frustration and also contribute to errors if the user did not realize that the toggle failed.

6.3.4 Two handed manipulation

This method definitely increases the intuitiveness when manipulating objects, but the technique is hard to master. The main problem is that both hands need to be visible within the HoloLens limited gesture frame when the interactions are taking place, which is can be hard to control when making anything but very small and slow gestures. It is also hard to get the rotation to work around every axis. As a workaround it is possible to lock certain axes, but that brings back the toggle method dilemma of having to switch mode for every interaction.

6.3.5 Voice commands

We use voice commands for quick and simple actions, such as creating objects or toggling visibility of menus. The efficiency of creating waypoints and cross line detection with voice was appreciated by many of our test participants. We have chosen to never use voice commands as the sole way of interacting with an object, in part because it can be problematic or even impossible to use efficiently in loud environment, but also, in quiet environments it might not be desirable and make the user feel uncomfortable. From the user testing we also found that some expressions are harder for the HoloLens to recognize or is heavily affected by different accents. For example, the voice command "Start" caused a lot of trouble for our Swedish testers, and we often asked them to use their best American accent if they got stuck - which solved their problem almost every time. We have noticed that when it comes to voice commands, personal preference also plays a big role. Some people were very reluctant to use voice control as a means of interaction, especially when there are other people around, while others were very happy to use their voice.

6.3.6 Using UI elements

Traditional UI elements have the benefit of being a recognized way to display information and to show available functionality. On the other hand, AR opens up for new interaction methods that might be less complex and more intuitive. We implemented a traditional menu system because we believed it to be the best way to show users new to AR what could be done within the application. We also used UI elements to adjust a camera's pan, tilt and zoom. From the user tests we could see that it was easy to understand the menus and the other UI elements, but we also realized that it could be slow and cumbersome. Compared to traditional computer setups with a mouse as input device, the air tap gesture used with the HoloLens is way less precise. Users would often miss their target, either resulting in no action or even the wrong action, which sometimes caused confusion. To

get away from clumsy menus and virtual UI elements we would have to try other ways to tell users what to do, such as highlighting interactable objects or give information when an object is gazed upon. We believe this will be a trade off between efficiency and how well a user understands a system, at least until people get more familiar with AR and some kind of standard for these things are agreed upon.

6.4 Future work

While working on the thesis, we naturally realized possible improvements and extensions to our application and the research area along the way. While they often were outside the scope of the project, or simply would take too much time to implement, a couple of them are still interesting to mention. In this section we list functionality we would have liked to add if we were to continue the development of our application.

Integrate with existing tools. It would be interesting to explore the possibility to integrate some of our solutions into existing tools that Axis uses today to sell and visualize their systems.

Adding textures to room scans, models and camera feeds. By generating textures based on what the HoloLens cameras capture, the virtual camera feeds and models can look more like the real world. This would greatly increase the feeling of realism in the application and give users an easier mapping between the real and virtual world.

Implement camera comparison tools. By generating camera feeds based on actual camera specifications, the virtual feeds could serve as a comparison tool for different camera models. This could help in choosing the proper model for different settings and scenarios (such as limited light, outdoor usage, facial recognition etc).

Adding collaboration functionality. The ability to share models and security systems generated in the HoloLens is something that is greatly desired. Implementing a way to collaborate between HoloLenses, via VR, desktops or even with tablets would be a significant step in this direction.

Expanding beyond cameras. A modern security system is made up of more than just cameras. Implementing ways to plan a more complete system, where the choice of parts and information about wiring in a building could influence each other would be a logical next step.

Improve functionality and design according to user feedback. A lot of good feedback was received in conjunction with the user tests. Much of this could be implemented given more time.

Chapter 7

Conclusion

This thesis has investigated if and how AR can be applied to the planning, implementation and visualization of security systems.

To answer these questions, an AR application for Microsoft HoloLens was developed. The application focused on scenarios where a user could implement some basic functionality in a camera-based security system. The system could then be verified with simulations of events by letting a virtual character move through the room where the system had been set up.

Even though it is still early in the development-cycle, our application has shown that there exists major advantages to using AR in the context of complex problem-solving and data presentation. Additionally, in talking to users and experts in the field, it is clear that many of the rough edges will disappear in the next few years.

A user study was conducted, and the scores from both SUS and NASA TLX exceeded our expectations. All the test sessions went very smoothly and all participants completed every task. This was somewhat surprising, since many of our users had little to no experience when it comes to AR. The concept of AR has made it into the mainstream conscious with Pokemon GO, and it really showed. Even though many users were amazed at first, nobody was really surprised by the holograms or the capabilities of the application.

Our tests did not capture enough data to draw conclusions about which of our versions performed best, especially since the test scores showed the users favoured Version 2, and the interviews indicated the opposite. More similar trials are required to determine if there is a clearly better version, or if personal preference always triumphs.

Bibliography

- [1] *A Short History of Git*. Accessed on May 2018. URL: <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>.
- [2] Lennart Alexandrie. *His idea and passion was turned into a world leading industry*. Sept. 2008. URL: <https://www.securityworldmarket.com/na/News/Interview-of-the-Month/his-idea-and-passion-was-turned-into-a-world-leading-industry>.
- [3] Patricia L Alfano and George F Michel. “Restricting the field of view: Perceptual and performance effects”. In: *Perceptual and motor skills* 70.1 (1990), pp. 35–45.
- [4] Apple. *ARKit*. Accessed on May 2018. URL: <https://developer.apple.com/arkit/>.
- [5] Ronald Azuma et al. “Recent advances in augmented reality”. In: *IEEE computer graphics and applications* 21.6 (2001), pp. 34–47.
- [6] Aaron Bangor, Philip T Kortum, and James T Miller. “An empirical evaluation of the system usability scale”. In: *Intl. Journal of Human–Computer Interaction* 24.6 (2008), pp. 574–594.
- [7] Egor Bogatov. *Xamarin/urho-samples*. Accessed on June 2018. URL: <https://github.com/xamarin/urho-samples>.
- [8] John Brooke et al. “SUS-A quick and dirty usability scale”. In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.
- [9] Per Christensson. *User Interface Definition*. Accessed on May 2018. 2009. URL: https://techterms.com/definition/user_interface.
- [10] Axis Communications. *About Axis - History*. 2015. URL: <https://www.axis.com/about-axis/history>.
- [11] High Tech Computer Corporation. *HTC Vive*. Accessed on May 2018. URL: <https://www.vive.com/eu/>.
- [12] Ditto.com. *Redefining Augmented Reality*. Accessed on April 2018. URL: <http://blog.ditto.com/2013/10/redefining-augmented-reality.html>.

- [13] *Git Large File Storage*. Accessed on February 2018. URL: <https://git-lfs.github.com/>.
- [14] Google. *ARCore Overview*. Accessed on May 2018. URL: <https://developers.google.com/ar/discover/>.
- [15] Rebecca A Grier. “How high is high? A meta-analysis of NASA-TLX global workload scores”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 59. 1. SAGE Publications Sage CA: Los Angeles, CA. 2015, pp. 1727–1731.
- [16] Sandra G Hart. *NASA Task Load Index (TLX): Volume 1.0; Paper and Pencil Package*. Accessed on May 2018. 1986. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000021488.pdf>.
- [17] Sandra G Hart. “NASA-task load index (NASA-TLX); 20 years later”. In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 50. 9. Sage Publications Sage CA: Los Angeles, CA. 2006, pp. 904–908.
- [18] Sandra G Hart and Lowell E Staveland. “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research”. In: *Advances in psychology*. Vol. 52. Elsevier, 1988, pp. 139–183.
- [19] Lunds Tekniska Högskola. *About EASE*. Accessed on April 2018. 2015. URL: <http://ease.cs.lth.se/about/>.
- [20] Andrew Hunt and Scott McGlashan. *Speech Recognition Grammar Specification Version 1.0*. Accessed on May 2018. 2004. URL: <https://www.w3.org/TR/speech-grammar/>.
- [21] Joseph J LaViola Jr et al. *3D user interfaces: Theory and practice*. Addison-Wesley Professional, 2017.
- [22] Meta. *Meta 2*. Accessed on May 2018. URL: <http://www.metavision.com/>.
- [23] Microsoft. *Cortana Home Assistant*. Accessed on May 2018. 2018. URL: <https://www.microsoft.com/en-us/cortana>.
- [24] Microsoft. *HoloLens Hardware Details*. 2018. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details>.
- [25] Microsoft. *Microsoft HoloLens*. 2018. URL: <https://www.microsoft.com/en-us/hololens>.
- [26] Microsoft. *Mixed Reality Toolkit - Unity*. Accessed on February 2018. URL: <https://github.com/Microsoft/MixedRealityToolkit-Unity>.
- [27] Microsoft. *Use gestures*. Accessed on May 2018. URL: <https://support.microsoft.com/en-us/help/12644/hololens-use-gestures>.
- [28] Microsoft. *Voice input*. Accessed on May 2018. 2018. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input>.
- [29] Microsoft. *Windows Mixed Reality*. Accessed on May 2018. URL: <https://www.microsoft.com/en-us/windows/windows-mixed-reality>.

- [30] Paul Milgram and Fumio Kishino. “A taxonomy of mixed reality visual displays”. In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), pp. 1321–1329.
- [31] Niantic. *Pokemon GO*. Accessed on June 2018. URL: <https://pokemongolive.com/en/>.
- [32] Donald A Norman and Stephen W Draper. *User centered system design: New perspectives on human-computer interaction*. CRC Press, 1986.
- [33] Occipital. *Structure Sensor*. Accessed on May 2018. 2018. URL: <https://structure.io/>.
- [34] Jason Odom. *Two hand manipulation coming to the HoloLens*. Accessed on Mars 2018. 2018. URL: <https://hololens.reality.news/news/dev-report-3d-live-tiles-two-handed-manipulation-more-are-finally-coming-hololens-0183241/>.
- [35] Erwin Peters et al. “Design for collaboration in mixed reality: Technical challenges and solutions”. In: *Games and Virtual Worlds for Serious Applications (VS-Games), 2016 8th International Conference on*. IEEE. 2016, pp. 1–7.
- [36] Jannick Rolland and Hong Hua. “Head-mounted display systems”. In: *Encyclopedia of optical engineering* (2005), pp. 1–13.
- [37] Jeff Sauro. *Can you change a standardized questionnaire?* Accessed on May 2018. 2016. URL: <https://measuringu.com/change-standardized/>.
- [38] Jeff Sauro. *Measuring usability with the system usability scale (SUS)*. Accessed on May 2018. 2011. URL: <https://measuringu.com/sus/>.
- [39] D Sharek. “NASA-TLX Online Tool (Version 0.6)[Internet Application]”. In: *Raleigh, NC. Retrieved from http://www.nasatlx.com Google Scholar* (2009).
- [40] David Sharek. “A useable, online NASA-TLX tool”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 55. 1. SAGE Publications Sage CA: Los Angeles, CA. 2011, pp. 1375–1379.
- [41] Alexandra Sheehan. *How These Retailers Use Augmented Reality to Enhance the Customer Experience*. Accessed on May 2018. URL: <https://www.shopify.com/retail/how-these-retailers-are-using-augmented-reality-to-enhance-the-customer-experience>.
- [42] Richard Stoakley, Matthew J Conway, and Randy Pausch. “Virtual reality on a WIM: interactive worlds in miniature”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co. 1995, pp. 265–272.
- [43] Linus Torvalds and Junio Hamano. *Git: Fast version control system*. Accessed on May 2018. 2010. URL: <http://git-scm.com>.
- [44] Microsoft Yoon Park. *Two Hand Manipulation & normal mode Bounding Box*. Accessed on Mars 2018. 2018. URL: https://github.com/Microsoft/MixedRealityToolkit-Unity/blob/master/Assets/HoloToolkit-Examples/Input/Readme/README_TwoHandManipulationTest.md.

- [45] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. “Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR”. In: *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2008, pp. 193–202.
- [46] Ning-Ning Zhou and Yu-Long Deng. “Virtual reality: A state-of-the-art survey”. In: *International Journal of Automation and Computing* 6.4 (2009), pp. 319–325.

Appendices

Appendix A

Personal information questionnaire

Personal information

*Required

Your test ID: *

Your answer

What's your gender? *

- Female
- Male
- Other
- Prefer not to say

What's your age? *

- 15-25
- 26-35
- 36-45
- 46-55
- 55+

How tall are you? (cm) *

Your answer



Are you right or left handed? *

- Right
- Left
- Both

Do you wear glasses? *

- Yes
- No

What's your experience with AR (Augmented Reality)? *

- I have never heard of AR
- I know what AR is
- I have read up on AR on my free time
- I have studied AR courses
- I have researched/worked with AR professionally
- Other:

What's your experience with AR applications? *

- I have never used an AR application
- I have used AR applications on my phone
- I have used AR applications with AR headsets/glasses
- I have used AR applications on my phone and with AR headsets/glasses

Have you ever used a Microsoft HoloLens? *

- Yes
- No
- No, but I have used other AR equipment

SUBMIT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Google Forms

Appendix B

Modified SUS questionnaire

Product evaluation form

*Required

Your test ID: *

Your answer

1. I think that I would like to use this application if I were to design a security camera system *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

2. I found the application unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

3. I thought the application was easy to use. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

4. Using this application again, I think I could go through the scenario without technical assistance *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree



5. I found the various functions in this application were well integrated. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

6. I thought there was too much inconsistency in this application. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

7. I would imagine that most people would learn to use this application very quickly. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

8. I found the application very cumbersome to use. *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

9. I felt overwhelmed using the application *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

10. I could see the relevance of the application *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Appendix C

Interview Questions

What did you think about the GUI, menus etc? How was the interaction?

(Version 1) What did you think about being able to place the UI elements where you wanted?

(Version 2) What did you think about the GUI elements followed you everywhere?

After using one, which do you imagine you would prefer?

What did you think about the camera placement?

How did the placement indicators help you, or were they helpful at all?

What did you think about the room scanning step?

Was it easy to understand what to do?

Do you have any general thoughts about a security system being visualized in Augmented Reality like this?

Can you see any limiting factors or opportunities?

Did you notice the model that was created after the scanning stage? Did you see that the cameras appeared in the miniature model as well?

Appendix D

Test scenario and user guide

Start the application:

The two first options are selected in the menu.
Press "Start".

Step 1 - Build a virtual model of the room you are in

In front of you, there's a button called "Main Button" or "MB".
Walk around in the room and look at walls and the floor to scan.
The ceiling does **not** need to be scanned.
You know that an area is finished when it has a green overlay.
When the entire room is scanned, you can move on by pressing the MB which now shows "Click here to finish scan".

Step 2 - Minimize the model you build in Step 1.

The program automatically fill areas that wasn't scanned.
When the program is finished, press MB which now shows "Build Model".
You can look at the model you have created. It is located somewhere next to you.

Step 3 - Place 3 cameras on appropriate places on the walls

There is a camera in front of you, above MB.
To place the camera; press on it once to make it movable. The camera will now follow your gaze. Press again to place it.
Choose an appropriate placement according to the program's suggestion.
To create the next camera; press the MB, now showing "Create Camera 2" and place it.
Repeat for the third camera.
Press the MB, now showing "Confirm Placement" when you feel satisfied with all three cameras.

Step 4 - Explore how the menu works

There is a menu in front of you.
The menu can be minimized by pressing the square above it.
The menu can be moved by dragging the square.
"Camera Menu" will help you with Step 5.
"Avatar Menu" will help you with Steps 6-8.

Step 5 - Move a camera so it records Bob

Move Bob (he's located at the floor close to you), so he is standing on the white marker.
The marker is found on the floor.
An active camera has blue color.
Up to your right you there is a display which shows the feed of the active camera.
To switch which camera is active, you can press on a white camera.
Move a camera so Bob is visible in the image (from which angle doesn't matter).
When the camera is in place, zoom in or out so Bob is clearly visible. You do this from the menu.

Step 6 - Construct a path for Bob to follow

The path is constructed by placing waypoints.

You create a waypoint by choosing in the menu, or by saying "New Point".

The waypoint is placed in the same way as the cameras.

Place three waypoints on the areas that is marked on the floor.

Step 7 - Place a "Cross line detection" along the path

Somewhere on the path you should place a line which will react if Bob is crossing it.

The line is marked on the floor.

You create a line by choosing in the menu, or by saying "New Line".

The line is placed in the same way as the cameras.

Step 8 - Simulate how the system works

Start a simulation by choosing in the menu or by saying "Start".

Follow the events.

Appendix E

Test plan

Usability test of AR prototype: Arriba

Purpose

The purpose is to gather data regarding an AR prototype created as a master thesis project at Axis Communications. The data will be used to evaluate the usability of the prototype. It will also be used as a means of user feedback to help with further development of the prototype.

Questions

1. If tag along or being able to move UI elements is preferred?
2. How does it feel to interact with “traditional” UI elements in Augmented Reality?
3. Was it easy to understand where cameras could be placed, and where their optimal placements were?
4. Did the camera preview give an understanding of how a camera feed would look like in the real world?
5. How did the room scanning and room mapping feel? Did it feel relevant? Was it hard/easy to map the room? Did the WiM give a good model of the real room?
6. How well can a security system be described/visualized in AR? Limiting factors?
7. Did the simulation of the system increase the understanding of how a security system could work?

Data

Both subjective and objective data will be gathered during the test sessions. However, most emphasis will be on subjective data. This is because the main goal of the experiment is to get an understanding of what users think about the usability of the prototype and what their thoughts are on the different interaction methods with the AR technology.

Tasks

Task	Sub tasks	Completed when...	Maximum time spent
<i>For users with zero experience with the HoloLens:</i> AR/HoloLens Introduction	1. Let the user play around with some HoloLens application, in order to master the “pinch” gesture.	User feels ready to move on	10 min

Initialize the model and create the WiM	<ol style="list-style-type: none"> 1. Scan the room (walls, ceiling, floor and objects) 2. Finalize the model mesh 3. Create the WiM 	The mesh is completely “sealed” and a WiM is visible	Must be finished. ~5 min
Get familiar with the UI	<ol style="list-style-type: none"> 1. Explore the different UI elements. Try to get a basic understanding of how it works and how to use it. 	The user feels ready to move on, or maximum time is reached	5 min
Place cameras	<ol style="list-style-type: none"> 1. Grab a camera 2. Place the active camera on a wall, preferably on a suggested spot (visualized with a green overlay) 3. Spawn a new camera. 4. Repeat step 1-3 until three cameras are placed in the room 	<p>Three cameras are placed in the room.</p> <p>The cameras should be visible in the WiM</p>	5 min
Interact with the cameras	<ol style="list-style-type: none"> 1. Select a camera that is not active. 2. Adjust the camera position by dragging it so that the avatar (Bob) is visible in the feed. 3. Zoom out with the camera, using the adjustment menu 	<p>The main test tasks has been fulfilled</p> <p>The user feels ready to move on, or maximum time is reached</p>	5 min
Set up a path for Bob to follow	<ol style="list-style-type: none"> 1. Place Bob at his starting position (user’s choice) 2. Place the existing waypoint as Bob’s first target to reach 3. Spawn an additional waypoint (by clicking the UI button, or by voice command), and place it as the next target in the path 4. Repeat step 3 until at least four waypoints has been placed 	When at least four waypoints have been placed in the room	5 min
Place cross line detection along the	<ol style="list-style-type: none"> 1. Place at least one cross line detection (by clicking the UI button, or by voice command) somewhere 	When at least one line has been placed along the path	5 min

path	on the path created with the waypoints		
Start the simulation	<ol style="list-style-type: none"> 1. Start the simulation 2. See if the system acts according to expectations 	When Bob has reached his target	1 min

Procedure

Formalities

The test person will be asked to fill out a form regarding personal information and previous AR experiences. A consent form will also be signed, allowing the research group to record the session and use the gathered data in publications.

Introduction

The test leader will explain the context of the experiment and go through the different parts of the scenario to prepare the test person for what's to come.

Preparation

A HoloLens introduction and hands-on will be given in order to learn the basic interactions and gestures, which will be used in the experiment.

Experiment

The test person will carry out the experiment. The test leader(s) will have a feed of the HoloLens and will assist if the user get stuck. The test persons will also be able to access the written scenario at any time if they need to refresh their memory.

Questionnaires

The users will fill out a (modified) SUS and a Nasa TLX (first part only) questionnaire when the experiment is done.

Interview

After the experiment a semi-structured interview will take place. The test person will be asked a couple of predefined questions and also be able to discuss the experiment more freely in the end.

Test Environment

The tests will be carried out at Axis Communications conference room Aantré if the user is an Axis employee. For other users the tests will be carried out at the LTH VR lab at IKDC. The experiment will be recorded, both using the HoloLens feed and a video camera. The real world test room will be prepared with indicators of how the avatar path is supposed to look like, and where it would be appropriate to have cross line detection.

There will be two slightly different applications when performing the experiment. Half of the users will be able to move their UI elements freely in the world, while the other half will have the UI discreetly follow them around automatically.

Test Participants

The test participants will primarily be chosen from within three groups:

1. Employees at Axis Communications with knowledge and understanding of camera security systems
2. Students at Lund University with knowledge and understanding of AR technology
3. Voluntary testers who wants to try our prototype, but without any prior experience with AR or security systems

Appendix F

Information and consent

INFORMATION and CONSENT

Please read this consent form carefully before you decide to take part in this study.

Background and purpose

We're writing a master thesis at Axis Communications where we research on how Augmented Reality (AR) can ease the planning of a security system and how one can visualize different requirements in security systems. Additionally, we are exploring and comparing various interaction models within AR.

How is the study carried out? The participant will go through a number of tasks. The session will take about 45 minutes.

What are the potential risks and benefits? The experiment is completely safe and does not involve any risk of injury. If you take part in the experiment you will get the opportunity to try out Microsoft HoloLens, one of the best AR glasses on the market.

Handling of data and privacy. Your answers and results will be stored under an anonymous code which only will be available to the research group. All the results of the study is presented combined, no individual patterns will be described on identifiable. No personal data or information will be saved. Audio and video will be recorded during the test and might be used internally.

Responsible researcher for this study at Lunds University is Günter Alce, +46 79-347 67 77, gunter.alce@design.lth.se

The study is voluntary. Your part in this study is completely voluntary. You can revoke your participation at any point during the experiment without any particular reason, and doing so without any consequences.

Consent (Keep this copy!)

I have read the form, and my questions regarding the extent and background of the study has been answered. I consent to participate in this study:

Signature

Date

Printed name

Consent (Give this copy to the test leader)

I have read the form, and my questions regarding the extent and background of the study has been answered. I consent to participate in this study:

Signature

Date

Printed name

Appendix G

Tutorial and briefing

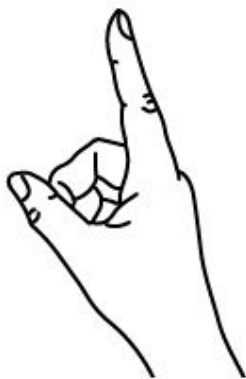
Intro:

Du jobbar med att planera säkerhetslösningar och skall testa ett nytt verktyg som med hjälp av Augmented Reality skall förenkla din process. En kund har gett dig ett antal krav, och du ska nu ta fram ett förslag på hur systemet ska se ut. Din arbetsprocess består av tre steg; modellering av rum, utplacering av kameror, och till sist finjustering och verifiering.

Inledning

För att du ska känna dig mer bekväm med HoloLensen och dess hologram, får du bekanta dig med några enkla interaktionsmönster. Applikationen innehåller två objekt som ligger framför dig; en kub, och en sfär.

I nuvarande version stödjer Hololensen två gester, "Air Tap" och "Bloom".



1. Finger in the ready position



2. Press finger down to tap or click

Kuben

För att byta färg på kuben, klicka på den.

För att flytta den, klicka på den, håll kvar samtidigt som du flyttar den.

Sfären

Klicka på sfären för att göra den flyttbar

Den kommer nu följa din blick

Rikta blicken där du vill placera den och klicka igen