Alexander Næsheim

Augmented reality based operator interface for increased depth perception and decision support during subsea IMR operations

Master's thesis in Industrial Cybernetics
Supervisor: Aksel Andreas Transeth
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Kunnskap for en bedre verden

Alexander Næsheim

Augmented reality based operator interface for increased depth perception and decision support during subsea IMR operations

Master's thesis in Industrial Cybernetics
Supervisor: Aksel Andreas Transeth
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Kunnskap for en bedre verden

Preface

This report is the culmination of my time as a student at the Norwegian University of Science and Technology and represents my master's thesis within Engineering Cybernetics (TTK4900) delivered the spring of 2019. The contents of this report are produced through independent development and research including regular meetings with my supervisor Aksel Andreas Transeth and his colleague Esten Ingar Grøtil at SINTEF Digital. They are both senior research scientists at the department of mathematics and cybernetics. Throughout the spring they have provided guidance, experience and interesting discussion in our regularly scheduled meetings every other week. When it comes to software's used, Unity and Vuforia play the central roles when developing the main contribution, however, Creo Parametric is also used for 3D CADing of 3D printed models and all the CADs are made based on self-though methods. Throughout the development within Unity and Vuforia, the C# programming language is used and the scripts are built from scratch with some inspiration from different sources in the community, and self-thought methods to achieve the desired functionality. A development with Python and the OpenCV library is also briefly mentioned in this report. This development was built from a foundation created by Tiziano Fiorenzani, which had the functionality to track one single 2D ArUco marker. Other than that, self-research, trial, and error, and learn by doing is what builds the foundation of this report and its investigation of the research question.

For the formalities at NTNU, Anastasios Lekkas has been my internal supervisor being available for guidance regarding formalities from NTNU's side. When it comes to what the university has provided me with in order to conduct the experiment and investigate the research question, see the list below:

- The department of engineering cybernetics provided me with a Dell Optiplex 9010 computer running Windows 10 and an office available every day at all hours. A Logitech Webcam C925e was also provided.
- A room to conduct the experiment was provided by the department of engineering cybernetics.
- To prepare the room for the experiment, cardboard, plywood, and wood beams were provided by the workshop at the department of design.
- Several parts were also 3D printed to be used in the experiment, the workshop at the department of engineering cybernetics could arrange a 3D printer for use by appointment. This workshop also provided tools to construct the complete experiment setup.
- Anastasios Lekkas also offered to provide a robotic arm to use in the experiment, it was, however, uncertainties around when this would be available other than it would be late in the semester because the arm was part of another master's thesis.

Acknowledgment

Research and development are done better when its discussed with others, and this report has benefited from engaging discussions with my supervisors *Aksel Andreas Transeth* and *Esten Ingar Grøtil*. I am very grateful of their will to share their experience and guidance throughout the work with this report. I would also like to thank my fellow graduate students for motivating and inspiring discussions, as well as those who participated in the experiment. They are all mentioned below. Finally, I want to thank my family for the love and support they have given me throughout my time as a student.

Thanks to the participants who partook in the experiment: *Anders Kjellevoll, Anders Stene, Edvard Hofstad, Julie Bjørke Andersen, Karianne Talset, Lars Kristian Gjelstad, Thomas Alexander Frekhaug, and Øystein Utbjoe.*

Alexander Næsheim

06.06.2019

Summary

When remotely operating a vehicle or drone through a controller where the only visual information of the controlled object is a video feed from its camera displayed on a 2D monitor, you are dealt with a task to control an object in a 3D environment by only receiving 2D directional information. This report looks at emerging technology within augmented reality (AR) to help provide increased depth perception (i.e., depth of field information) to operators, operating under these conditions. In addition to providing depth perception, this report also seeks out to introduce decision support capabilities to the operator interface in order to aid operators during decision taking situations.

On subsea fields at the seabed, equipment is seldom within human reach and are therefore interacted with though Remotely Operated Vehicles (ROVs). Frequent ROV operations are those falling in under the IMR category which includes inspection, maintenance, and repair operations. This is a scenario like the one mention above where the operators operating the ROVs usually only have 2D information available. With this in mind, this report seeks out to answer the research question: *Can utilization of emerging technology within AR help introduce usable depth perception and decision support capabilities to a 2D monitor based interface between subsea ROVs and their operators?*

To investigate the research question an AR operator interface is developed in Unity together with the Vuforia developer kit. Different depth tools are designed and implemented with the intention to provide the operator with depth information, while other AR components are also included to presents relevant operation information in an attempt to create decision support capabilities. When developing this AR system one of the greatest challenges turned out to be the alignment of AR objects onto the physical world.

After the AR system was designed and developed, the capabilities mentioned in the research question was evaluated through a designed, arranged and conducted experiment. In this experiment, 8 participants had to complete three different operations while operating a wooden beam which could be maneuvered with 5 DOF. This wooden beam called "the tool" has a camera attached to it, and the participants first had to complete the three operations by only having the pure video feed from that camera available at a 2D monitor. In the second run, they had the AR system presented on the monitor. After the operations, the participants answered a survey to evaluate general aspects and the usability of the designed, developed and implemented system.

From the above, it was concluded that the developed operator interface had obtained depth perception and decisions support capabilities through the implemented AR components. In addition, based on the usability evaluation, the system was rated to have good usability. There were also discovered areas which could benefit from improvements and further work to increase robustness and reliability, as well as the overall usability before this kind of system could be utilized in the interface between subsea ROVs and their operators.

Sammendrag

Droner eller kjøretøy styres ofte ved hjelp av kontrollere i områder som ikke er til syne for operatøren. Disser er derfor vanligvis utstyr med et kamera som filmer og presenterer denne filmen på en 2D skjerm, slik at operatøren kan se omgivelsene. I slike situasjoner står operatøren ovenfor oppgaven om å måtte styre noe som befinner seg i et 3D miljø med bare 2D informasjon tilgjengelig. Denne rapporten ser på hvordan augmentert virkelighet (AR) kan brukes for å gi operatører dybde syn, altså 3D informasjon, når de opererer under slike forhold. I tillegg er det også av interesse å se på mulighetene for å kunne lage beslutningsstøttende funksjonalitet ved hjelp av AR, å hjelpe operatører til å ta bedre beslutninger.

Subsea felt er plassert i områder hvor det er ekstremt vanskelig og farlig for mennesker å befinne seg. Derfor brukes fjernstyrte kjøretøy når fysiske oppgaver skal utføres på subsea utsty. Slike oppgaver faller ofte innenfor kategorien IMR operasjoner som er inspeksjons-, vedlikeholds- og reparasjonsoperasjoner. Under slike operasjoner oppstår situasjonen beskrevet ovenfor hvor de som styrer undervannsfartøyene (ROVene) ofte bare har 2D informasjon tilgjengelig. Derfor ønsker denne rapporten å se nærmere på problemstillingen: *Er det mulig ved hjelp av fremtredende teknologi innenfor AR, å utvikle brukervennlig dybde syns og beslutningsstøttende funksjonalitet i et 2D monitor basert grensesnittet mellom subsea ROVer og dem som styrer de?*

For å undersøke denne problemstillingen har et AR operatør grensesnitt blitt utviklet ved hjelp av Unity og utviklerpakken Vuforia. I dette grensesnittet er det implementert ulike dybde verktøy, med hensikt om å presentere dybde informasjon til operatøren. I tillegg er det også utviklet andre AR komponenter som viser relevant operasjonsinformasjon, med ønsket om å tilby beslutningsstøttende funksjonalitet. Det ble møtt ulike utfordringer i prosessen med å utvikle dette AR-systemet, hvor kanskje den største var oppgaven med å legge AR informasjonen oppå den fysiske virkeligheten på en robust og god måte.

For å kunne evaluere om det utviklede grensesnittet var velfungerende, ble det satt opp et eksperiment for å teste AR-systemet. I dette eksperimentet deltok 8 frivillige personer. Til eksperimentet ble de det utviklet et verktøy. Dette kan styres med 5 DOF, og er utstyrt med et kamera. Ved å kontrollere dette verktøyet skulle deltakerne gjennomføre tre ulike operasjoner. Disse ble utført to ganger. Første gang bare med hjelp av video strømningen fra kameraet, som ble presentert på en 2D skjerm. Andre gangen skulle det samme gjøres, bare nå med AR-systemet presentert på skjermen. Etter disse operasjonene svarte deltakerne på en spørreundersøkelse for å evaluere ulike områder ved systemet og brukervennligheten av det.

Rapporten konkluderer med at det ved hjelp av AR er mulig å utvikle et system som inneholder dybde syns og beslutningsstøttende funksjonalitet, selv om det presenteres på en 2D skjerm. Gjennom test og evaluering kom det i tillegg fram at AR grensesnittet, med den ønskede funksjonaliteten, har god brukervennlighet. Det ble også i rapporten oppdaget og diskutert komponenter som gjerne ikke utnyttet sitt fulle potensiale. Disse kunne ha tjent på forbedringer og økt fokus under videre arbeid for å øke robustheten, påliteligheten, og

den overordene brukervennligheten av systemet. Dette, i tillegg til fokus på å løse de begrensningen som rapporten presenterer, må arbeides videre med før et slikt AR grensesnitt kan tas i bruk for å assistere operatører som styrer subsea ROVer.

Contents

List of Figures

List of Tables

Nomenclature

e.g. = **exempil gratia**, for example
etc. = **et cetera**
i.e. = **id est**, that is
ROV = **Remotely Operated Vehicle**
DOF = **Degrees Of Freedom**
O&G = **Oil and gas**
hp = **Horsepower**
Practitioner = **The person overseeing the experiment**
SA = **Situation awareness**
Pose = **Combination of position and orientation**
API = **Application Programming Interface**
Fps = **Frames per second**
UI = **User interface**
SD = **Standard deviation**

The developed system within this report will be referred to as the AR operator interface and AR system. The report will also describe the users of this system as operators when thinking of this developed system used in a subsea IMR operation scenario, participants when speaking of the users during the experiment and user is used when mentioning the system in an arbitrary user case.

Chapter 1

Introduction

As an introduction to this report both the background and motivation for why the topics of this report are important and actual is presented. Following comes a description of what this report sets out to investigate and answer, as well as the contributions created along the way. The chapter is finished off by introducing the outline of this report.

1.1 Background and motivation

During the last half decade, the world has witnessed an enormous drop in the oil price. In mid-summer 2014, the price for a barrel of oil was 106 US dollars. From that point on and about one year and a half later in the start of 2016, the oil price had reached a record low of under 30$ per barrel of brunt crude [1]. Luckily this only marked the bottom and was a turning point. Since then the price has risen and today, in 2019, we can see an oil price that is somewhat stabilizing around 70$ a barrel, even though it is a very varying marked. This is visually displayed in Fig. 1.1 which shows the oil price over the last decade.

Naturally, the shifting prices above took its toll on the O&G industry which back then was characterized by unhealthy cost structures [2]. However, when the drop came, it forced the companies to focus more on increasing their capital discipline, productivity improvements and utilization of new technology [3]. With the Norwegian Petroleum Directorate's forecast in 2019 showing that "... oil and gas production will increase from 2020 and up to 2023. Overall production will then approach the record year of 2004" [4]. Indicates that the tides have turned and the situation has started to stabilize and companies are experiencing increased activity out on the fields again.

Figure 1.1: A graph showing the oil price over the last decade. Data and graph provided by Macrotrends [1].

This crisis forced the companies to think in new ways and change to accommodate the lower selling price of their product. For an industry that was characterized by great returns because of extremely high oil prices from 2011 to 2014, this was not a straight forward task for the companies and many, sadly, did not manage to survive. However, those who did had to focus on increasing capital discipline and productivity improvements to reduce operation time and cost. A way to do this appeared to be increased utilization of new technology which over the long term would make their portfolios more profitable against low break-even prices on investments [3]. One of the great stories emerging from this whole situation is Equinor's achievement of cutting their break-even price from 100$ to 27$ per barrel [2]. With this kind of capital discipline, today's oil price of about 70 US dollars must result in great margins.

With the above in mind, the subsea industry is increasingly pushing towards reducing cost and operation time with increased safety in subsea inspection, maintenance, and repair (IMR) operations. SINTEF together with industry-leading partners (IKM, Oceaneering, TecnipFMC, and Equinor), The Research Council of Norway and NTNU work together in a project named SEAVENTION to meet this industry goal by researching underwater robot autonomy for these subsea IMR operations [5]. A step towards achieving autonomous Remotely Operated Vehicles (ROVs) is the development of systems which can support the ROV operators during these kinds of operations. Today, the operators operating ROVs do so by looking at several 2D monitors with operational information and a live video feed from the camera on the ROV at the seabed. This essentially means that the ROV operators in most cases have to operate the ROV in a three dimensional (3D) environment by looking at a two dimensional (2D) monitor. This leaves out the depth of field information, meaning that operators have to operate large ROVs with very limited depth information. The participants of the SEAVENTION project, which this report is a part of with focus area shown in Fig

1.2, have had the gratitude of watching such operations live at Oceaneering's HQ and mission control center in Stavanger, Norway. Here we witnessed what must've been the most frustrating thing ever, gripping a handlebar with a gripping manipulator without 3D vision. The ROV operator used five attempts to even reach the right position in the depth field to grip the handlebar. Immediately after seeing this the motivation for developing a tool or system to provide some degree of depth perception for these ROV operators was set. As a stepping stone towards full robot autonomy in subsea operations, it must be possible with today's technology to provide some type of depth information to these operators. They are doing crucial operations which the subsea production is dependent on in order to function properly. It is important to know that humans can't touch subsea equipment while installed in the production and processing system, meaning that humans only means of interaction whit these installations are through ROVs and thereby states the importance of a great and informative interface between these two.

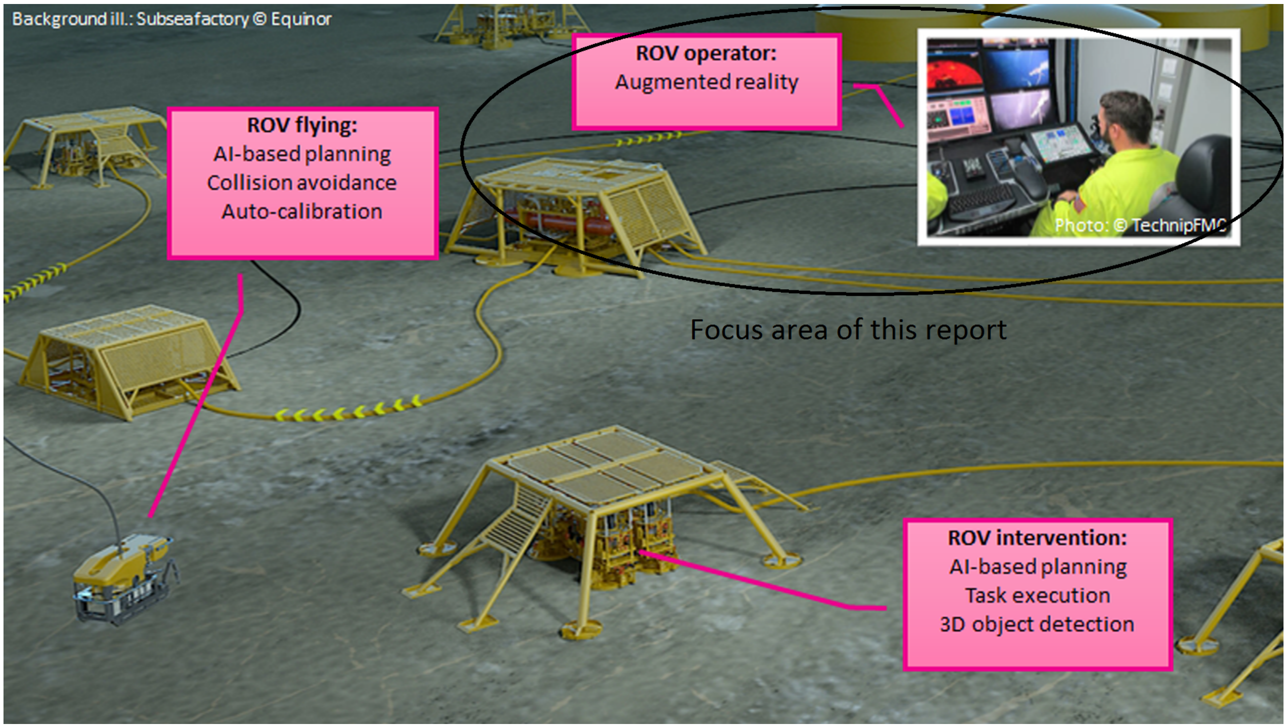


Figure 1.2: Illustration of this reports focus area within the SEAVENTION project, source: [5].

These subsea IMR operations are vital for the survival of subsea fields and wrongdoings and wrong decisions during such operations could potentially cost companies and stakeholders a substantial amount of money. There has been done much research looking into methods to increase safety and making these operations easier to execute. What most of this research have in common is that they often look at how to improve the ROV or how to make them more versatile. This includes research on modeling the dynamics of the ROV [6, 7], wireless control of the ROV from top side computers [8], ROV control system to automate the installation of valve trees [9], and development of new tools to make the ROVs tasks during subsea IMR operations easier [10]. This and more subsea ROV research not mentioned look at important aspects of the ROV. However, it also introduces the impression that not so much research has been focused on approaches to make the actual aspect of operating the ROVs easier for the operators. Therefore, with what appears to be an alternative angle to approach the discussion of improving subsea ROV operations, this report seeks out to find ways to improve the quality of the experience of operating an ROV. Following the above, this report, together with SEAVENTION's motivation of improving man-machine operations, will look into the potential of providing both depth perception and decision support to the operator interface displayed on 2D monitors [5, 11]. This could potentially be possible with the help of emerging technologies within augmented reality. Such development would hopefully create great value for both the operators and the broad specter of stakeholders dependent on ROV executed subsea IMR operations.

1.2 Goal and method

The superior question arising during the starting phases of the work connected to this report, was "How can AR help subsea ROV operators during IMR operations?", after research and experiences as a partaker in the SEAVENTION project a more specific goal and research question for this report emerged, where depth perception and decision support had central roles.

The more specific goal of this report is to investigate how the implementation of augmented reality can provide increased depth perception for subsea ROV operators during IMR operations. Introducing AR to an operator interface opens the door for integrating a wide range of operational information to the interface system. A wider goal is, therefore, to also look at the potential of equipping an operator interface with decision support capabilities. The intention of this is to deliver a supportive feeling towards the operators when they interact with subsea equipment through ROVs at the seabed. From this goal and the superior question emerges a research question which is to be investigated and answered within the lines of the scope of this report. The scope will be presented in the next section.

Research question: *Can utilization of emerging technology within AR help introduce usable depth perception and decision support capabilities to a 2D monitor based interface between subsea ROVs and their operators?*

This research question is investigated by developing and implementing an AR operator interface which is supposed to provide both depth perception and decision support to its users. To test if these capabilities actually are usable and function as intended, an experiment is designed and arranged. The experiment is conducted with the help of volunteering participants. After they have used and experienced the AR operator interface, their impressions and feedback are collected through a substantial evaluation highlighting different usability goals and general aspects of the developed and implemented AR system. Lastly, are this feedback presented and analyzed in order to answer the research question.

1.3 Scope

This report is a part of a bigger research project called "SEAVENTION" [5]. Within this project, the report is intended to look into how AR can improve the interface between ROV operators and the actual ROV, an illustration is shown in Fig. 1.2. This is done by looking at methods within AR to, in a usable manner, integrate depth perception and decision support capabilities to this interface.

Because other teams within the SEAVENTION project are looking into regular and 3D camera sensors and computer vision to enable robust object tracking subsea, this report will be limited to marker-based AR to direct more focus towards development of relevant AR information. Other than that will this report stand on its own feet, with no further ties to the SEAVENTION project.

Based on the above, is the scope of this report to implement some AR system with marker-based tracking that functions in air. This part will take inspiration from our previous work in [12]. Also included in the scope is a non-automated experiment setup with different operations designed to test the functionality of the system. This is within the scope with the intention to initiate work towards a long-term goal, outside of this scope, to implement such a system between subsea ROVs and their operators. It is, therefore, not in the scope to actually test this system with or within direct ties to subsea ROVs and IMR operations, however, these will be mentioned and some discussed as the system is designed with the intention to one day function as an interface in the subsea industry.

1.4 Contributions

The main contribution of this report is the developed AR operator interface with decision support and depth perception capabilities. It highlights the potential rewards of utilizing emerging computer science technology to improve important aspects of the interface between human operators and mechanically and cybernetically advanced ROVs. More specifically this report contributes with a system that through marker-based AR technology is capable of providing its user with “in real-time” depth information. This is done without the need for additional expensive projection equipment, as this system presents 3D information directly onto a regular 2D monitor. Such a system can be used in situations where a vehicle or drone at a distance and out of vision is supposed to be remotely operated by looking at a 2D video feed coming from that operated object. Undeniably, it should be easier to navigate a vehicle or drone in a 3D environment with 3D information available, compared to only having two-dimensional information at the disposal.

The contribution of using AR on operator interfaces is not only limited to presenting depth information, it rather opens the door for a huge spectrum of potential information to be augmented into the operator’s vision of the physical reality. One category of this is decision support, which the contributed AR system also emphasizes to provide. This means that this developed and implemented contribution sets out to reduce human errors based on uneducated and doubtful decisions, which could lead to more effective and efficient operations.

Other contributions are the design of an experiment to test decision support and depth perception capabilities of marker-based AR systems. The main component enabling us to do this is the in house designed and 3D printed ball joint which lets the participants control the object bearing the camera in the experiment with 5 DOF. This contribution could open the door for more frequent testing, as the experiment setup on its own (without the camera and AR system) is analog. There is, therefore, no need for advanced and complex robotic arms to test changes and additions to the developed AR system.

1.5 Outline of report

This report is divided into six different chapters and an appendix. Combined they serve the purpose of describing the work done to create an AR operator interface, the process done to test the created system, and the analysis done to enable the conclusion to answer the research question presented in the introduction.

- **Chapter 1** - *Introduction*: The first chapter gives an introduction by presenting the motivation, goal and contributions of this report.
- **Chapter 2** - *Background and theory*: In this chapter, three wide main topics augmented reality, subsea technology and usability are defined and described, these represent the topics in which this report will be based around.
- **Chapter 3** - *Developments and implementations*: After the theoretical foundation is gained, software tools used to develop the AR operator interface are introduced together with the process of implementing this AR system.
- **Chapter 4** - *The design, arrangement, and conduction of the experiment*: As the AR system now has been created, this chapter describes how an experiment to test the system is designed, arranged and conducted.
- **Chapter 5** - *Analysis*: This is the largest chapter of the report. Here implementation results and both quantitative and qualitative experiment results are presented and discussed. These results are divided into different groups and results are presented and discussed group wise throughout this chapter.
- **Chapter 6** - *Conclusion and further work*: From the analysis in the previous chapter a conclusion and recommendations for further work emerges and is presented in this chapter.
- **Appendix** - The appendix is divided into four groups. Appendix A contains the handouts given to the experiment participants, appendix B consist of the drawn 3D models which is 3D printed and used in the experiment, appendix C has all the C# scripts used in Unity, while appendix D have the Python OpenCV ArUco marker tracker script.

Throughout this report, especially in chapter 4 and 5, there are some figures that are videos. To start these simply click on the figure. In order for them to run, you have to open the PDF file in an updated version of Adobe Acrobat Reader (not in a web browser extension). Inside Adobe Reader you might be asked to activate 3D content, this message would appear in a yellow banner in the header, this feature should be allowed. When "always allowed" please exit the document and re-open it again if videos won't play. If the report is not read inside Adobe Reader, the videos are also included in the enclosure marked with their figure name.

It is recommended to read this report inside an updated version of Adobe Acrobat Reader, to run the videos inside this report.

Chapter 2

Background and theory

In chapter 2 we introduce the theory which this report is based on. In order to investigate the research question theory within Augmented Reality is needed to make the basis for the implementation done. Some theory about Subsea Technology is also needed to understand how ROVs are used and why they are used. Lastly, usability will be introduced to enable proper evaluation of the implementation in the hope of collecting results which can be analyzed and hopefully help answer the research question. The goal of this section is not to dwell on about the different subjects, but rather give å short introduction to the subjects and provide hopefully necessary background information relevant for this report.

2.1 Augmented Reality - a brief introduction

Recent technological developments are changing the way humans experience physical and virtual environments. More specifically is augmented reality (AR) enhancing our view of the physical reality. In order to define AR, one could look at The Oxford Dictionary's definition of augment which is "to make something greater by adding to it". With this definition one could say that AR is a technology which overlays useful computer-generated data on the user's vision of the physical world, to create a reality that is enhanced or augmented. Typically, this AR technology can be observed through two main observation interfaces. The first one is through a head-mounted display which offers an egocentric viewpoint, which is when information is projected directly to the user's senses, in this case, the eyes [13]. The second is an interface where the user views the AR on 2D monitors or screens (e.g., smartphones, tablets, TV's, etc). The latter makes for a window-on-the-world viewpoint, which was introduced by Kristian et al. [14].

AR technology is essentially 2D or 3D objects, layered on top of the real world, seen through an observational interface. Same as for the interfaces there are two types of methods used when adding the AR layers to enhance reality. One of them is see-through technology, in this case, AR objects are projected onto a direct view of the real world. The second is video-based technology where the real world is captured by a camera and its video feed is mixed with virtual elements and then presented to the user through an observational interface [15]. The information presented through AR could be purely synthetic, such as from a computer simulation, or it could be copies of real-world information represented digitally. The information can be presented in different fashions, one of the methods is statically, such as a still digital photograph and 2D graphics as in Fig. 2.1, and another is through static or dynamic 3D digital graphics and models such as in Fig. 2.2 [16].

Figure 2.1: 2D objects integrated in the real world through video recording on a phone. Source: [17].

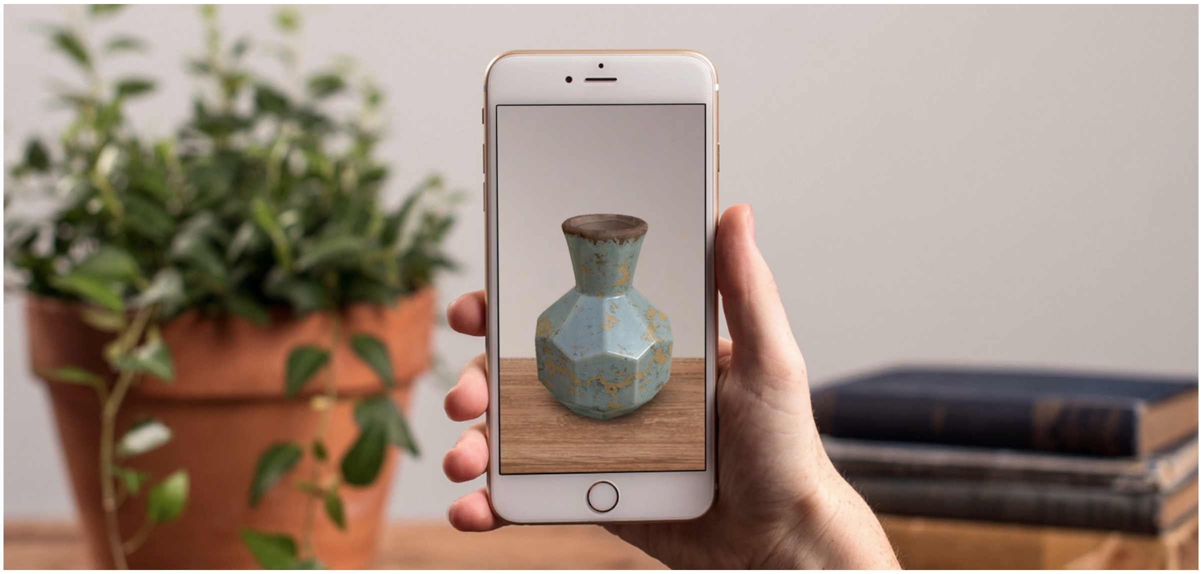


Figure 2.2: A 3D object integrated in the real world through video recording on a phone. Source: [18].

One of the greatest challenges within AR is to ensure that the inserted graphics and effects are precisely integrated and do not jitter or drift about in the observational interface as the user moves around and changes perspective. This is particularly challenging when the user moves his/her head rapidly around when using head-mounted displays. The same goes for movement of the camera when having a windows-on-the-world viewpoint. The system then has to predict and estimate the new location very fast and accurately in order to present the AR information in a stable manner [15]. In order to layer this information onto the real physical world, an AR-system is dependent on a technique to calculate the relative 3D dimensions of the reality around it. There are different methods to acquire this information, one could for instance use information coming from the Gyroscope in an iPhone. In this report, however, we are utilizing computer vision based methods to collect this information.

Computer vision-based tracking

Within computer vision tracking there are two main potential ways to go, either marker-based or markerless tracking. When using marker-based tracking, there has to be at least one physical marker in the scene captured by the camera. The computer vision system is trained to track markers and by doing so it calculates the position of the camera relative to a marker. From this, the center of the world is placed in the center of the marker and the computer vision system can use this as a reference to orient itself after. At the same time, this could be flipped around by having the camera act as the center of the world and the markers tracked by the camera be objects floating around in the captured scene. In order for the marked-based system to have something to add AR effects to, these markers have to be created or chosen. There are a few principles to follow when deciding which to use. First of all, you want objects with high contrast in your markers (e.g., black and white). Secondly, you want these objects to have a certain uniqueness to set them apart from different objects found in common scenarios, you also want them to contain some distinct corners. Finally, the markers should include objects with some rotation or outline to distinct the objects orientation, this helps the AR program recognize orientation and adjust the AR effects accordingly [19]. What all these principles have in common is that they together describe a marker that has a lot of features in it, and a feature-rich image is simply an image with a lot of prominent parts and characteristics. Features is, therefore, components of an image which makes it unique compared to others.

The other tracking method, called markerless tracking also strives to find a center point to augment the reality relative to. Meanwhile, they differ from each other in the way they find this point. Markerless systems do not rely on predefined physical objects. Instead it seeks to do what the computer vision system is programmed to do, which could for instance be segmenting the captured scene. If the scene is segmented by color, the system could layer every red component in the scene with yellow AR objects. Another example is to recognize faces which can be used to add animated effects around or directly on the detected head. This concept is widely used by Snapchat [20]. Based on this, markerless tracking is the more versatile of the two methods, but marker-based often provides more reliable tracking. And since markerless, in most cases, is the more complicated to develop, marker-based tracking has seen the most popularity when it comes to AR development [21].

In this report, the Vuforia SDK will be used. It is an SDK which utilizes marker tracking techniques to layer AR information. It will be used with the Unity 3D engine to create a complete AR experience. These tools will be introduced in the next chapter. But before we leave the AR subject, the remaining of this section will present a brief look at the history of AR and describe the increased interest in AR from the industry.

A look at the history of AR

The first example of an attempt to add additional data to an experience was through a device called Sensorama, see Fig. 2.3. The cinematographer Morten Heiling created in 1959 this device which while seated inside the user could experience deep visuals, sounds, vibrations, and smells. Jumping ahead to 1975, Myron Krueger invented a device, the Videoplace, where the user could interact and manipulate the information and environments played on a screen. This was possible through the combination of a projection system and video cameras which produced shadows on a screen. It was, however, not before 1990 the term "Augmented Reality" was introduced by Boeing researcher Tom Caudell. Following the introduction of the concept of AR, the first properly functioning AR system came to light in 1992. At the USAF Armstrong's Research Lab, Louis Rosenberg developed an incredibly complex robotic system, called Virtual Fixtures displayed in Fig. 2.4. This system was capable of overlaying sensor information in workers work-environment, something that improved human productivity on the production line [12]. Since then there have been many other breakthroughs, some of the most notable includes:

- A mobile outdoor AR game named ARQuake developed by Bruce Thomas in the year of 2000
- The introduction of ARToolkit, a design tool made available in Adobe flash in 2009
- The introduction of the open beta "Google Glass" by Google in 2013
- Microsoft announcing their new augmented reality headset "HoloLens" and their exciting vision for AR in 2015. In 2019 they also introduced their updated and improved Hololens version, the "Hololens 2".

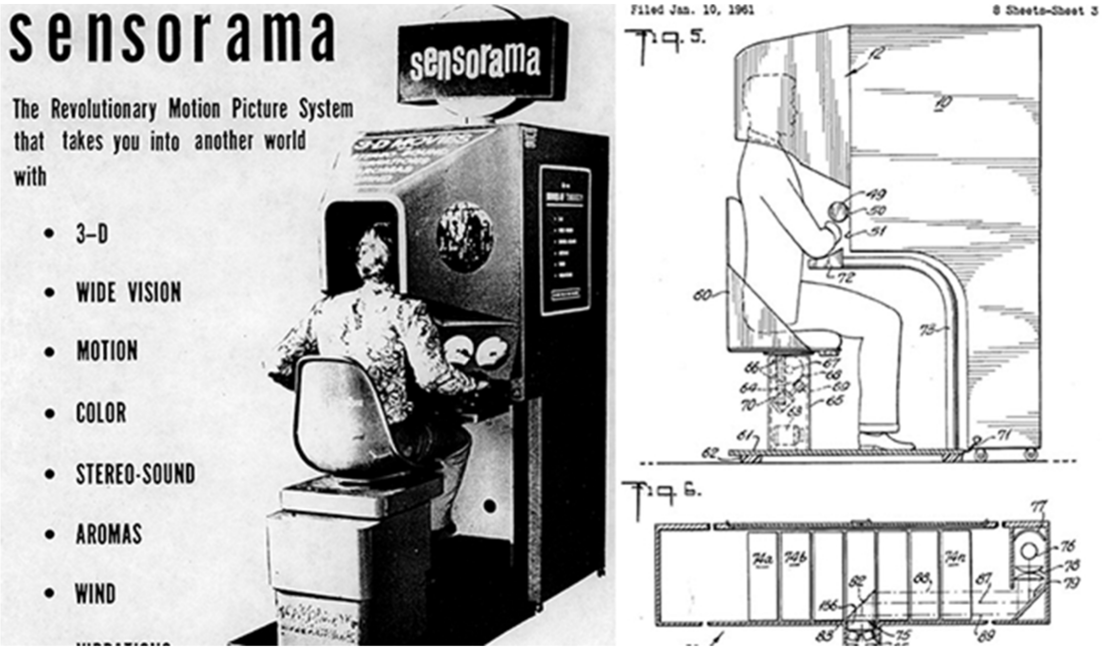


Figure 2.3: Illustration of the Sensorama device, invented by Morton Heiling. Source: [22].

Figure 2.4: The first properly functioning AR system, the Virtual fixtures, developed by Louis Rosenberg. Source: [23].

AR in industry

Moving up the timeline to today where different studies have shown that AR in different industry use-cases, such as assembly and maintenance tasks, can improve the overall performance through reduction in errors and operation time [24]. This has spiked an interest within different industries and the application of AR in the industry are thus increasing. Some of the recent applications are DHL's use of smart glasses for picking assistance within logistics [25], VW's AR-based visualization of manual instructions during maintenance (illustrated in Fig. 2.5) [26], and remote support from engineers or experienced operators through AR [27]. Due to little experience about the transferability of AR-based systems to real industrial applications, are these applications mostly used as prototypes in field tests. Most of today's applications are therefore driven by large companies with budgets to invest in new technologies. AR content is also usually created by specialists or third parties. For them, it is not always easy to incorporate non-declarative knowledge and adapt to rapidly changing methods and environments. This poses a general challenge for the implementation of AR in large scale industrial applications [24].

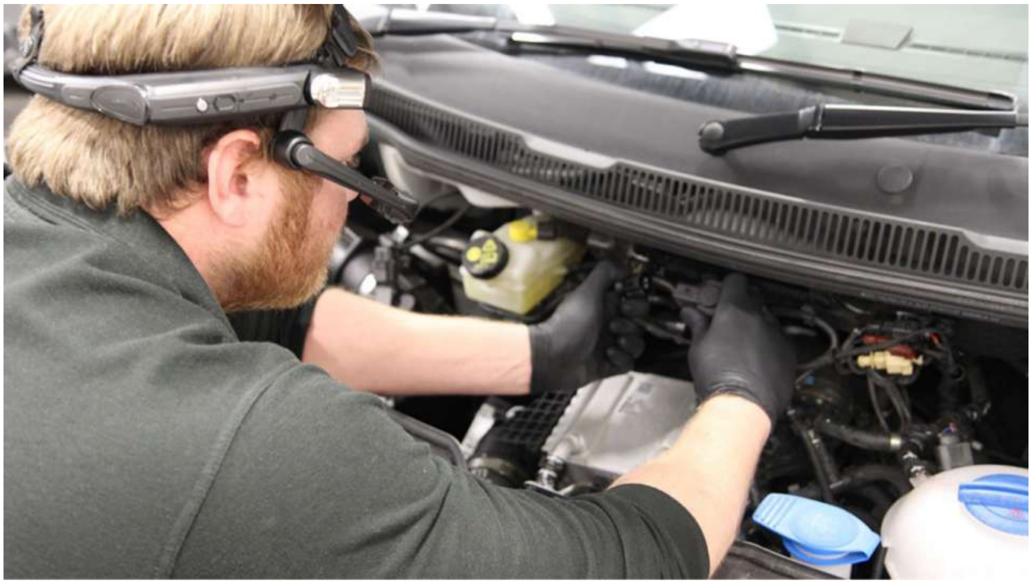


Figure 2.5: An operator using a head-mounted AR headset to see the AR visualization of the manual instructions during maintenance. Source: [28].

2.2 Subsea Technology - Oil and Gas

On a standard oil and gas platform, the oil and gas are sucked up from the reservoir to above sea level and aboard the platform. Here, the natural resources go through the platforms production facilities to be prepared for transportation to shore and a refinery. Imagine shorting the suctioning distance of O&G to the seabed and not several meters above sea level. This is essentially what subsea technology does. It moves the production and processing facilities on the platforms down to the seabed, or at least the most essential equipment. By doing this one can achieve increased recovery of hydrocarbons, open up the possibility to produce from marginal fields (i.e., fields that are difficult to produce from with a standard O&G platform), and increase production from already existing wells. These are only some of the benefits, together with cost savings, of subsea fields [29]. An arbitrary subsea field is illustrated in Fig. 2.6.

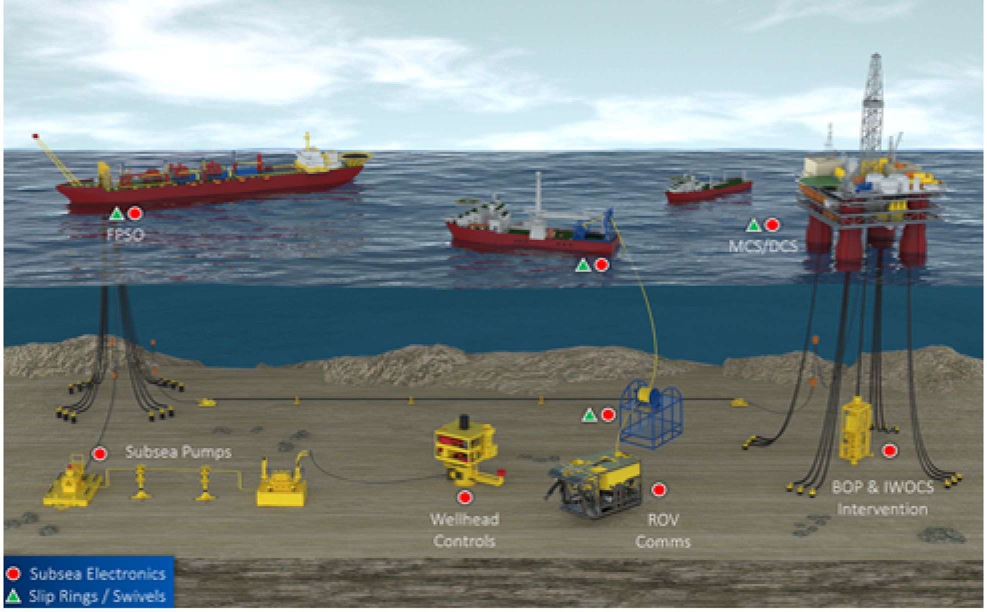


Figure 2.6: An arbitrary subsea field, the yellow components are the subsea equipment, source:[30].

Generally, not every component used topside for production could be brought down to the seabed. But in recent years more advanced production and processing systems have been installed subsea. This opens the door for separation, boosting and injection features directly at the seabed. These developments are often referred to as subsea factories, a term introduced by Statoil (Equinor) in 2012 during UTC in Bergen. Today they are expected to realize their vision of subsea factories within 2020 [31] and thereby "develop an offshore oilfield without any rigs visible on the surface at all" [32].

To create a subsea production and processing facility subsea one is very much in need of a method to do maintenance and repairs. Because these fields often are out of justifiable human diver conditions, the need for Remotely Operated Vehicles (ROVs) emerges. That is why ROVs are widely used during subsea operations.

2.2.1 Subsea IMR operations and ROVs

Before looking more into the ROVs, the IMR operations are first introduced. These are general activities performed on the subsea structures found at the seafloor. The abbreviation IMR can be divided up in three different categories, these are (I)Inspection, (M)Maintenance, and (R)Repair. These categories also represent the complexity of the operations. Where inspection has least complexity, as these operations are often scheduled condition monitoring operations which potentially could reveal the need for more complex operations. If the complexity moves one step up, it is categorized as maintenance. These operations are often actions needed to be taken based on earlier condition monitoring or from operators reporting about decreased performance. Lastly comes repair operations which are described by NORSOK Z-008 as "failures which requires immediate action towards cessation of performing the function, even though actual operation can continue for a short period of time" [33]. In other words, are IMR operations performed to maintain a sustainable flow of oil and gas from subsea wells, i.e., operations are done to make sure everything in the subsea system runs smoothly [29].

Subsea wells can be placed at depths as deep as 3000m below sea level [32]. Remotely Operated Vehicles (ROVs) are therefore used to interact with subsea equipment. There are mainly two different types of ROVs, inspection and work class ROVs. As its name indicates, inspection ROVs are used for visual inspections enabled by their cameras. The work class ROVs are much larger vehicles and one is displayed in Fig. 2.7. From the figure, one can see how they have two manipulators. One of them has the purpose of gripping on to handlebars on subsea structures to keep the ROV somewhat stable during specific tasks. The right one is more advanced, it has several degrees of freedom to help carry out the actual task and operations [34]. The controller used to maneuver the more advanced manipulator is shown in Fig. 2.8.

Figure 2.7: A work class ROV from Oceaneering [35] (the Millennium Plus), with the basic left manipulator and the more advanced right one.

The work class ROVs is usually used for maintenance and repair tasks, and the one illustrated above in Fig. 2.7 is so big that it weighs 4 metric tons while it is over 3 meters long and almost 2 meters wide and high. With this weight, the maximum thrust power generated by the propellers to move the vehicle is up to 900kg. It even has two hydraulic power units on board delivering 110 hp to control the manipulator arms [35]. These ROVs are usually operated in a control room on a vessel at sea level. From the vessel goes an umbilical down to the ROV. This is a cable containing all the communication components, electronics, and hydraulics. With this connection between the vessel and ROV, the subsea ROV operators can operate the ROV from their control room. A typical control room is illustrated in Fig. 2.9, where operators utilize different monitors with operational data as well as a big monitor with the video feed from the ROV. These work class ROVs also have to be versatile workers in order to cope with different IMR operations. This versatility is gained by using different tools. Some of the most used include: Cleaning tools, cutting tools, intervention tools, hot stabs and torque tools. [36, 29].

Figure 2.8: A typical controller used to control the right manipulator of a ROV like the one showed in Fig. 2.7. Source: [37].

Figure 2.9: Illustration of an on vessel work class ROV control room. Operators the controller and monitors with operation data and video feed to help them. Source: [38].

2.3 Usability of a system - human interaction

Usability is a wide and ambiguous term which refers to the quality of a users experience when interacting with products or systems [39]. This includes how information is accessed, ease of use, requirements to use it, how outcomes are achieved, etc. The usability is therefore influenced by a variety of factors that varies together with the intended purpose of the system. With all these factors it is important to realize that usability is not a single one-dimensional property, but rather a multi varying. To better control the factors and dimensions that affect the user's experience and ability to use the system, different usability goals are often defined. How these usability goals are defined and which aspects they highlight will be presented in the following subsection.

But first, when designing and developing systems, usability has a tendency to be seen as a luxury. Making something that is "pretty" and engaging is simply not enough. It must allow the user to produce the desired outcome in a convenient and relevant manner. This is more complex than just counting numbers of clicks or steps to achieve the desired outcome. To put this in context, Tom Stewart a consultant at System Concepts uses this example in talks "A user interface which shows last week's lotto winning number has to get you there in a few clicks to be "usable" (and tell you that you have not won - again). But if the system provided next week's winning numbers (a remarkable feat) then users could be persuaded to wear protective clothing, to hit the keyboard with a hammer to make it work and go through 100 clicks and would still be satisfied with the outcome" [40]. This quote enlightens how we generally are more prepared to put effort into learning how to use something if what it does is of great value to us. Looking at this enlightenment, the process of evaluating the usability of a designed and developed system must involve looking at the users, their goals, tasks, and environments[40].

2.3.1 Usability goals

Usability goals describe different aspects of the usability of a system. They were introduced to reduce the ambiguity associated with the terms "usability" or "user-friendliness". They also serve the purpose of defining these terms through measurable components. Because of this ambiguity, usability goals are differently defined. The ISO-9241-210 standard for user-centered design [41], for example, highlights effectiveness, efficiency, and satisfaction as their goals. Other examples are the interesting inclusion of learnability in [42], workload in [43] and situational awareness in [44]. The mentioned references do not solely define their usability goals as the once listed (i.e., learnability, workload and situational awareness), but they are part of a greater list of usability goals. In addition to the usability goals mentioned above, decision support [45] is added in this report because of its relevance to the research question presented in section 1.2. In the following paragraphs are the mentioned usability goals further defined [46, 47].

Effectiveness

In short, effectiveness says something about the accuracy and completeness with which users of a system achieve specified goals [41]. It looks at to-what extent the system enables the users to do the work they need to do. Because of this, effectiveness is a measurement of how good a system is at doing what it is supposed to do. This usability goal has, therefore, an overall view of the system, as it intends to measure how well the system performs [48, 49].

Efficiency

While effectiveness looking at the performance of the system, efficiency looks at the resources expended in order to let users achieve their goals accurately and completely. Focus is therefore shifted to look at what way the system supports users in carrying out their tasks. To achieve great efficiency you want your system to enable you to do the right things in order to achieve the goal. This could, for example, be a system that enables the user to carry out common tasks through a minimal number of steps. An inefficient system would, on the other hand, take the user through a detour of unnecessary steps in order to achieve common tasks [42, 50].

Satisfaction

When looking at the usability of a system, the user's freedom from discomfort while using the system is of great importance. This together with the potential of creating positive attitudes towards the use of the system is what the satisfaction usability goal is all about. The importance of creating a satisfying user experience should not be underestimated because of the potential positive momentum this could lead to (e.g., better working conditions, and increased motivation to use and learn the system, etc.) [41].

Workload

When talking about workload as a usability goal, vigilance plays a significant role. This role is described by Craig Donald in his chapter in [51]. He defines vigilance as follows "Vigilance refers to a capacity for sustained effective attention when monitoring a situation or display for critical signals, conditions or events to which the observer must respond" [51]. Based on this definition he's view is that high workload "can lead to reductions in vigilance as the person struggles to maintain accuracy and judgment under information and time pressure" [51]. At the same time does he believe that a very low workload may result in the operator being under-stimulated. This could lead to reduced vigilance, mental disengagement, and boredom. Similarly is Sanders and MComerick's [52] view that high workload could result in the user making incorrect decisions that could lead to disastrous situations such as an airplane crash. And the contrary, low workload may result in the user not monitoring the system properly and thus loses situation awareness [43]. These two views on workload underscore the importance of evaluating the potential workload the user could face using a system. By doing this, the chance for routinely over or under load the users during usage decreases, while the general usability should increase.

SA - Situation Awareness
Situation awareness can be described as the users internal model of the world around him/her at any point in time. It is within this internal model the decision on what to do and how to do it, is taken. This is graphically illustrated in Fig. 2.10 and one can see how SA is vital when taking actions. This is because, when elements in systems reduce our capability to percept and sens our surroundings it affects our ability to decide on and take actions. Workload, as introduced above, is also an important factor as it could reduce SA and interfere with the decision process. This process and SA is also influenced by the user's individual ability, experience and potentially training. To make way for great usability, it is a systems task to ensure that the user has great awareness and information flow while using the system.

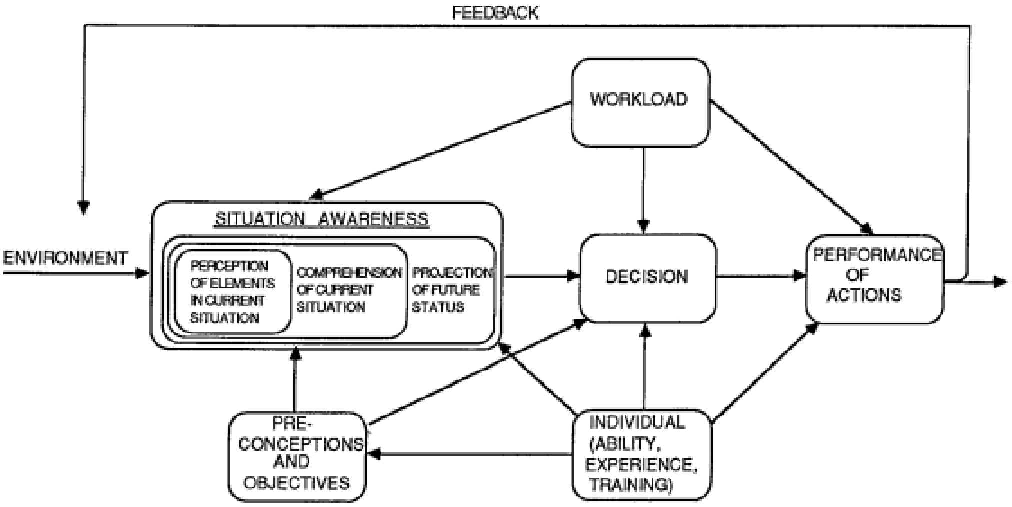


Figure 2.10: A graphical model showing the vital role of situation awareness in decision making. Source [44].

There exist different methods for measuring situation awareness. One way to do it is implicitly by correlate SA with something that is measurable. The number of collisions is measurable, and if an arbitrary use case has a high amount of collisions or near collisions this is expected to correlate with low SA. Thus, with this correlation one can implicitly measure the SA of a system. Another method is to explicitly measure the SA by pausing the user while using the system, to quiz him/her about their knowledge and information about their situation and location [46].

Learnability
Learnability looks at how easy a system is to learn to use. With a focus on this goal, the developers aim to reduce the hassle of learning how to use a system. Most people want to become competent and start doing tasks right away. This goes for interactive products and systems which is intended for everyday use (e.g., smart TV, mobile phones, email, video conferencing). To some degree, most people are prepared to use more time on learning complex systems with broader functionality (e.g., systems used at work, editing and

processing software, etc.). In these situations, there are often tutorials available online or in a manual to get started with the systems. These are however often very general and is therefore in most cases found tedious and often difficult to relate to the desired task people want to accomplish. Consequently, a key concern is to keep in mind how much time users prepare to spend learning a system. In worst cases, when not keeping learnability in mind, developed systems might have advanced functionality most users never will be competent enough to utilize or prepared to spend time learning how to use [42].

Decision support
When creating a system, one of the main goals of that development is often based around helping the user in doing something or enable them to do something they could not do before. If the development helps it such a way, it creates value for the user or other stakeholders. This is where decision support fits in as an important aspect of a system. It means just what it says, it is the process used to enable decision makers to make an informed decision which better utilizes resources and improve mission effectiveness. In practice, this could be achieved by combining information, tools, and techniques to understand the environment and analyze how and why previous things happened. These analyses can further be used to tailor a system which can evaluate and communicate projections, alternatives and potential impacts of different decisions, and present these in an informative manner to the user [45].
To measure the extent to which a system is supporting the user in making decisions, a method is to do so through quizzing the user about the felt support during usage. This results in subjective feedback. Another method is to measure the effectiveness with stepwise added decision support features, to get an indication on whether or not the system helps improve mission effectiveness [53].

2.3.2 Usability evaluation

When conducting usability evaluations, the focus is directed towards how well users can learn and use a system to achieve their goals. At the same time, you want to look at how satisfied users are with that process. This information is gathered by practitioners through a variety of methods that collects feedback from users about an existing system or plans related to a system in development [54]. Therefore, must usability evaluations be a part of the project plan when planning to develop a system or product aimed towards human users. This is emphasized by the ISO standard for user-centered design (ISO 9241-210 [41]), where usability evaluation is part of five key activities in the development phase. These activities are displayed in Fig. 2.11 and shows how the developing phase is an iterative cycle of key activities that includes evaluations.

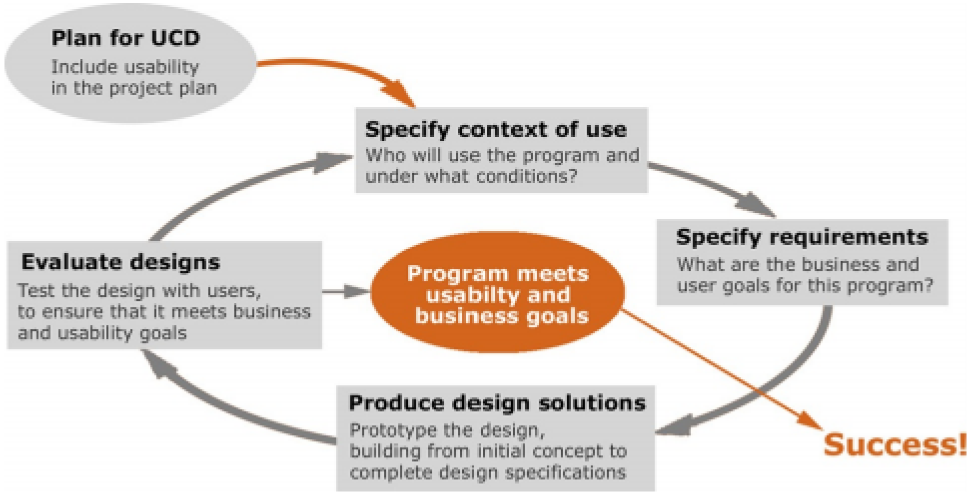


Figure 2.11: Five key activities in a system or product development project. Source: Original figure from [41], however, this is a re-designed version by [55].

During usability evaluations, there are two important things to keep in mind. The first is that evaluation must be integrated into the overall process because results are better interpreted and used when the designers and developers are able to observe and contribute to the work. Secondly, evaluation should occur early and often during the design and development phase. This is because the knowledge and insights from evaluations are cumulative, with each one building on the last. Integration of these two points in the planning phase could help a long way in achieving a greater user experience and to a greater extent help develop a system that performs within the defined usability goals [55]. Evaluations of usability could, therefore, be looked at as an evaluation that looks to collect information from users to evaluate how the system delivers within the defined goals. This information could be collected through several different methods. Some of these are focus groups, SUS, surveys, usability testing, first click testing, etc [54].

2.3.3 SUS - System Usability Scale

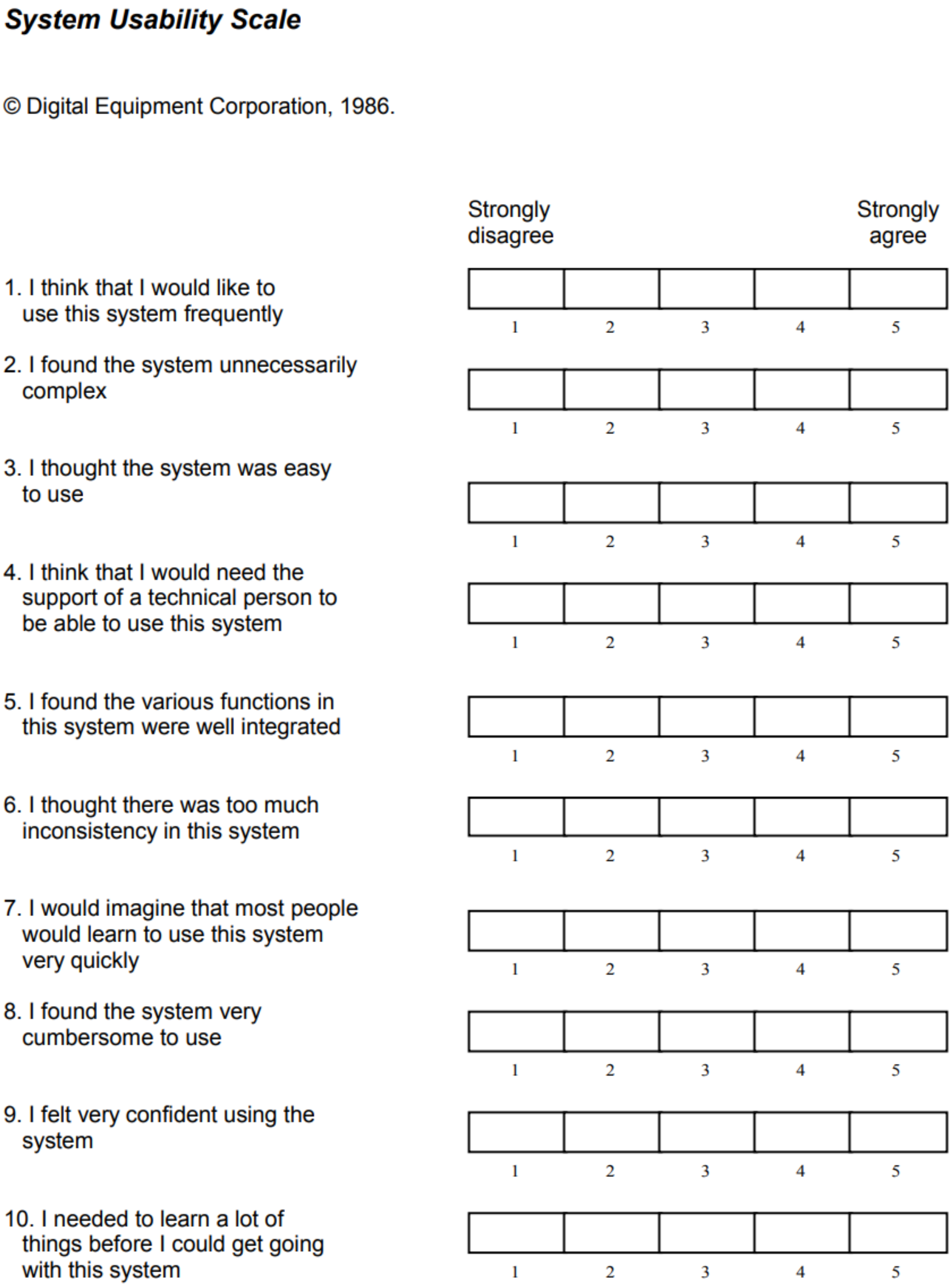
For over 30-years ago the System Usability Scale (SUS) was developed by John Brooke to subjectively measure usability of products and systems [56]. This questionnaire asks users to rate their level of agreement with 10 different questions covering a variety of usability characteristics. The level of agreement is measured through five response options ranging from strongly agree to strongly disagree. This is done by checking the boxes on the SUS questionnaire displayed in Fig. 2.12.
SUS has the main goal to quickly collect the user's immediate response after using a system or product and thus assess the usability without the need for complicated analysis [57]. The benefit of using it is, therefore, the fact that it can generate reliable results even on small sample sizes and its quality to effectively differentiate between usable and unusable system. This is one of the reasons why using SUS has become an industry standard with over 1300 references in articles and publications [58].

In order to take advantage of these benefits, one has to be aware of the somewhat complex scoring system when interpreting the scores. To prevent response biases caused by respondents not having to think about each question, the questions are asked with alternating positive and negative items. Because of this, the participants scores for each question have to be converted to a new number. This conversion is done by:

- For questions asked with odd numbers (i.e., question 1,3,5,7,9): subtract one from the participants response, which is rated on the scale shown at the right side in Fig.2.12.
- For the remaining questions (even numbered): subtract the participants response from 5
- As a result of the two points above, all values are now on a scale from 0 to 4 (with four being the most positive response)
- Finally, all the converted responses are added and multiplied by 2.5 to convert the total range score from 0 - 40 to 0 - 100. Even though the resulting scale ranges from 0 to 100, the scores should not be interpreted as percentages [56, 59].

Since the publication of SUS in 1986, different people have contributed with additional aspects to this usability scale. One of these contributions is Lewis and Sauro's addition of a usability and learnability scale. These additional scales have reasonable reliability and correlate highly with the overall SUS score. These additions offer practitioner to extract additional information from their SUS data at no extra cost other than some simple calculations [60]. Usability is calculated by adding the sum of questions 1,2,3,5,6,7,8 and 9, while learnability is calculated from the remaining questions (i.e., 4 and 10). Then, to make these two scores comparable with the overall SUS score (ranging from 0 to 100) the summed scores are multiplied with 3.125 and 12.5, respectively [61].

After scores are converted to the 0 to 100 scale, they can be interpreted, analyzed and compared to different SUS results. One method to interpreter the scores is through Bangor, Kortum, and Miller's [62] method of evaluating products and system in terms of adjectives such as "good", "poor" or "excellent". By analyzing the relationship between SUS scores and these terms of adjectives, they found a close correlation and thus proposed the possibility of taking the SUS score for a particular product and give it an adjective score. Another method uses a useful analogy to convey the mean SUS score to the traditional school grading scale (i.e., 100-90 = A, 89-80 = B, etc). This offers a scale that is familiar to most people and thus makes it easier for them to relate to [62]. The last tool is a set of acceptability ranges that could help practitioners determine if a given SUS score indicates an acceptable interface or not [63]. This tool is presented together with the other two methods of interpreting SUS scores in Fig. 2.13.



System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
2. I found the system unnecessarily complex	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
3. I thought the system was easy to use	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
4. I think that I would need the support of a technical person to be able to use this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
5. I found the various functions in this system were well integrated	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
6. I thought there was too much inconsistency in this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
7. I would imagine that most people would learn to use this system very quickly	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
8. I found the system very cumbersome to use	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
9. I felt very confident using the system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
10. I needed to learn a lot of things before I could get going with this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

Figure 2.12: The SUS questionnaire with the likert scale ranging from 1 to 5. Source [56].

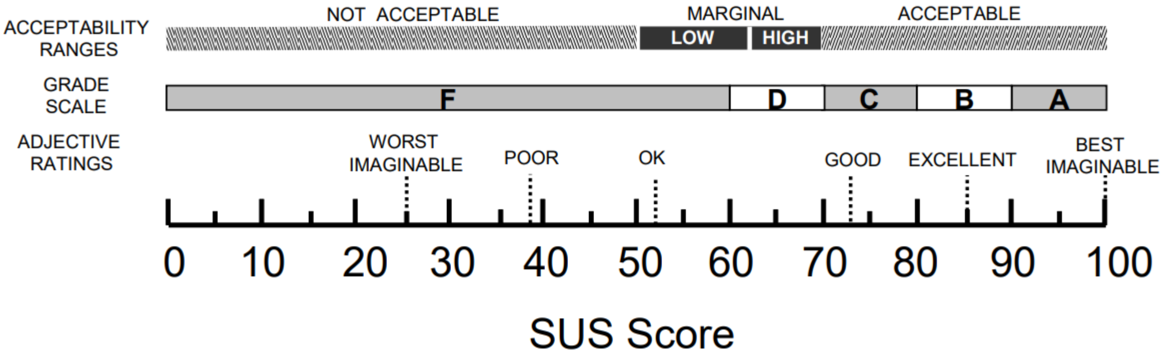


Figure 2.13: A comparison of the adjective ratings, acceptability scores for a SUS, and traditional school grading scales, in relation to the average SUS score [63].

Chapter 3

Developments and implementations

To investigate the research question of this report, some developments and implementations are done. The main implementation is the development of the AR operator interface. Before this implementation is presented, the chosen tools used are described followed by a description of the decision process behind that choice.

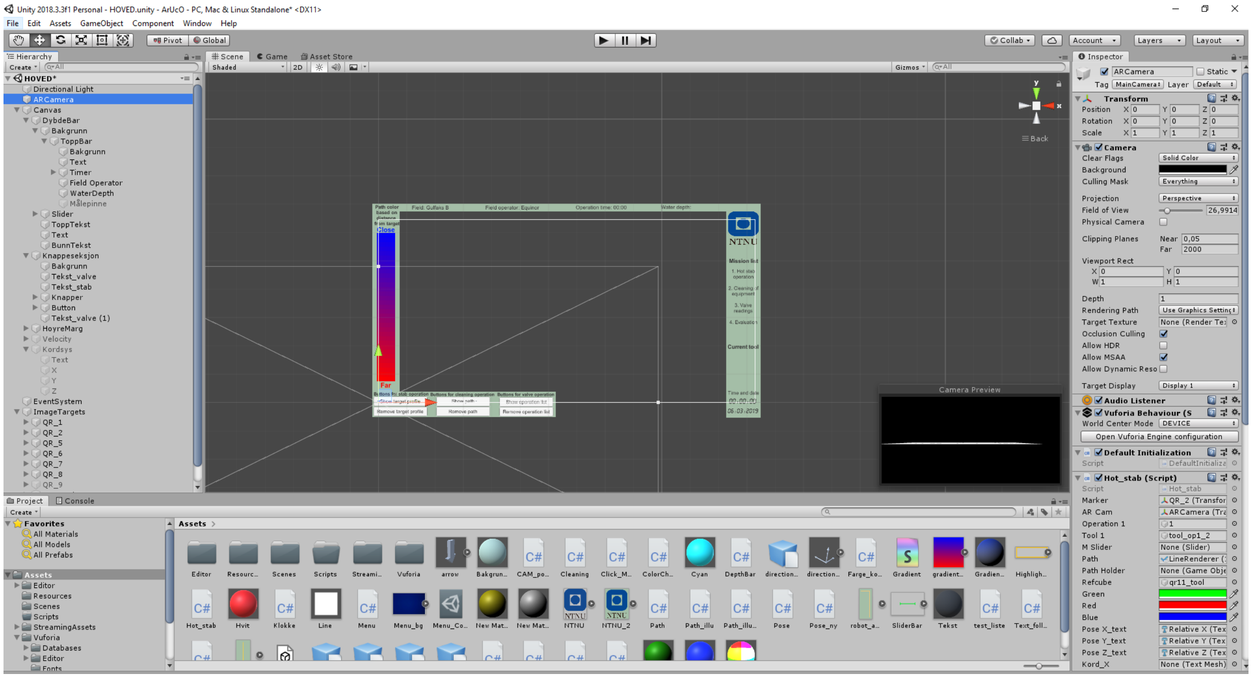
3.1 Software tools used

The main two software tools used when developing the main contribution, i.e., the AR operator interface, is Unity and Vuforia, these are described in this section. In addition, PTC Creo Parametric 4 was used to create 3D models for use in Unity and for 3D printing. The processing power behind these applications is the computer provided by the faculty and a Surface Pro 4 computer.

Unity
Originally Unity was a game engine, however with increased support for various VR, AR and MR platforms it could be considered a cross-platform tool. It is created by Unity Technologies and aims to help developers create games and applications. Unity is also recommended by Microsoft for HoloLens developers, something which validates its AR capabilities [64].

One of Unity's great strengths is their versatility when it comes to which platforms they can deploy to. It can deploy to Windows, OS X, iOS, Android, Web plugin, Flash, Xbox 360, PlayStation 3, and Wii U. With this versatility in mind, which their name originates from, it is not surprising that they are the most popular game engine with over one million developers. Another strength is Unity's ability to run the 3D graphics inside the program without deploying it to any device. This saves much time during development, as developers can see how the 3D graphics will look and how interactions with them will appear simply by running the created scene in Unity [21].

Unity has its own editor which lets the user create scenes by dragging and dropping objects in the editor, this editor is displayed in the Unity interface presented in Fig. 3.1. One can also implement additional logic such as object tracking or user input with JavaScript, C# or Boo scripts. In this report C# will be used as Vuforia uses C# for its scripts, thus it should be easier to communicate between them by using C#. In addition, C# is the language most professional studios use, together with the largest portion of the developer community. These C# scripts are developed in Visual Studios 2017 which integrates seamlessly with Unity [65]. Throughout the entirety of the developing process, Unity is not updated, as early experiences proved that updates could potentially mess up previous work and introduce bugs. Thus last years version 2018.3.3f1 is used for the entirety of the project.



☑ ARCamera ☐ Static
Tag MainCamera Layer Default

☑ Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ Camera
Clear Flags: Solid Color
Culling Mask: Everything
Projection: Perspective
Field of View: 26,9914
Physical Camera ☐
Clipping Planes Near 0,05 Far 2000
Viewport Rect X 0 Y 0 W 1 H 1
Depth 1
Rendering Path: Use Graphics Settings
Target Texture: None (Render Te
Occlusion Culling ☑
Allow HDR ☐
Allow MSAA ☑
Allow Dynamic Reso ☐
Target Display: Display 1

☑ Audio Listener

☑ Vuforia Behaviour (S
World Center Mode: DEVICE
Open Vuforia Engine configuration

☑ Default Initialization
Script: DefaultInitializa

☑ Hot_stab (Script)
Script: Hot_stab
Marker: QR_2 (Transfor
AR Cam: ARCamera (Tra
Operation 1: 1
Tool 1: tool_op1_2
M Slider: None (Slider)
Path: LineRenderer (
Path Holder: None (Game Obje
Refcube: qr11_tool
Green
Red
Blue
Pose X_text: Relative X (Tex
Pose Y_text: Relative Y (Tex
Pose Z_text: Relative Z (Tex
Kord_X: None (Text Mesh)

Figure 3.1: The Unity interface with its scene hierarchy on the left side, inspector at the right, assets at the bottom and the main scene editor in the middle.

Vuforia

Vuforia is one of the fastest marker-based tracking algorithms, it is an AR developer kit originally created by Qualcomm but was acquired by PTC in 2015 [66]. It is now also integrated, as a part of Unity [67]. The developer kit is available for free which makes it widely used and it, therefore, has an active community with many daily users on their forums.

Probably the main tool that Vuforia provides is the ARCamera, which brings the web camera video feed from the computer into Unity, ready for developers to integrate layers of AR objects onto it. At the same time is this component responsible for detecting markers in the scene. These markers are integrated into the Vuforia engine by uploading them to their database. When markers are uploaded to the database and detected by the camera, the ARCamera will calculate information about the orientation and position of the camera in relation to the detected marker. In Vuforia these uploaded markers are called "Image Targets". These can be arbitrary images uploaded to the database. Inside the database, these are used to train the engine to detect them. By following the marker tips provided in section 2.1, best tracking results are achieved. In addition, Vuforia provides some documentation on how to estimate the minimum size that the target should have. This is done by dividing the distance between the marker and camera by 10. The equation to calculate this is shown in Eq. 3.1.

$$Estimate\ of\ min.\ marker\ size = \frac{Distance\ between\ marker\ and\ camera}{10} \tag{3.1}$$

For example, in a scenario with camera to marker distance of 1 meter, the estimated minimum marker size is $\frac{100cm}{10} = 10cm$ [68]. After the image is uploaded one can essentially go to Unity and add the AR content desired to be displayed when the ARCamera discovers the marker, or "Image Target" [69]. An example of an AR object placed on top of an Image Target is shown in Fig. 3.2. The way the ARCamera manages to find and track these targets is by storing tracking data of the images in datasets. This tracking data is characteristical information about that specific image, i.e., information about edges and contrasting areas in the image. While the ARCamera processes the live video feed, it searches for areas that match the data in the stored datasets. If a match is found, the corresponding image to that specific dataset is thus found by the camera in the scene, and the AR object/objects are layered onto it. The ARCamera can search for up to 100 different images in the scene in real-time, something that really showcases the power of the Vuforia engine [21].

Figure 3.2: An image uploaded to the Vuforia database, which then acts as an image target (marker) with an added AR object on top of it. Source: [69].

3.2 Deciding what software tools to use when developing the AR system

To find these tools presented above to develop a system which gives the end user 3D information on a 2D monitor is not a straight forward process. This is because of the significant variation in complexity and functionality of the software, developer kits and also the operating systems. This process of deciding on what tools to be used should not be underestimated as the choice of tools could potentially make or break a project [70]. With this in mind, different types of tools are tested and evaluated before the main developing process can take place. In this section, the evaluation process is described and explained with the intention to reason why the different tools are used.

The initial tools tested was a setup with ROS running on the Ubuntu operating system. This foundation offers great customization and developing possibilities. However, this setup requires scripts in C++ which the author of this report is unfamiliar with. To, therefore, reduce the amount of C++ coding needed to develop an AR application, methods to bridge ROS with Python OpenCV and Unity was tested and researched. The great flexibility this kind of setup potentially could offer was a huge motivational factor when trying to figure out the communication between the different tools mentioned. After a lot of trial and error, it was decided to try a different setup, as the goal to develop a functioning AR system with the desired capabilities felt out of reach at that point. There were simply too many potential bottlenecks to be fixed or worked around, communication bridges to be built, and time-consuming challenges such as acquiring sufficient knowledge and experience with C++, Ubuntu, and ROS before developing could start [71, 72, 73, 74, 75, 76, 77, 78, 79].

After understanding the complexity of combining the tool mention above, the gained knowledge and experience was used to try to develop using Python OpenCV and Unity in Windows. Because, Unity is not directly supported on Ubuntu, running the development on Windows should solve some of the problems faced earlier. The idea of developing a tracking system in OpenCV and use Unity to design the AR information was an intriguing thought. With the help of the ArUco library, marker-based tracking is possible with python OpenCV. To test the capabilities of tracking with ArUco markers, a camera calibration script and tracking script is developed. The initial script with the camera calibration parameters resulted in robust tracking, but only of one marker. This functionality was provided by a template, developed by Tiziano Fiorenzani [80]. The template was then improved to track multiple trackers. At this time it was also interesting to test if it was possible to create 3D ArUco markers and track them. To test this the script was further improved, a cube was 3D printed, and ArUco markers were attached to each of its sides. Using cube geometry it was possible to track the center of the 3D marker, as seen in Fig. 3.3, but it was not as robust as the single marker. This might be caused by several factors one of which is the fact that each of the ArUco markers was not perfectly aligned on each of the sides of the cube.

Figure 3.3: The developed OpenCV program which tracks multiple ArUco markers, both 2D and the developed 3D marker.

Even though some progress had been made with the development of a functioning tracking program, the communication between OpenCV and Unity was still troublesome. One of the difficulties was the fact that the amount of applications that can run the webcam at the same time is limited to one in Windows. In this setup, both OpenCV and Unity require a live video feed from the webcam to function properly. Surely, there are solutions such as streaming the video feed to one of the applications [81], however, this could reduce frame rate and increase delays in the system which is not ideal when tracking objects to estimate their position. Another of the many difficulties was the fact that Unity does not support python, which makes the paring of OpenCV and Unity somewhat more difficult[82]. A method to integrate OpenCV and Unity by using DLL files was successfully achieved following the instructions of Thomas Mountainborn in [83, 84]. However, this solution requires both C++ OpenCV and C++ scripting in Unity which even though could be time-consuming to learn, should be less time consuming on Windows rather than having ROS and Ubuntu in the mix as well. It is worth mentioning that after the integration was up an running, a substantial problem accrued. The ArUco library did no longer function as before, and after some troubleshooting with no positive outcome, the motivation to develop with Mountainborn's integration was heavily reduced. Another solution would be to purchase an OpenCV for Unity asset to create Unity scripts with OpenCV in C# [85]. This would solve much of the communication problems between OpenCV and Unity, and make it possible to use the ArUco library [86] for tracking and pose estimation, but again, this asset was definitely not free with a fee of 100 US dollars.

The reason why Unity was a part of the previous two alternative setups is due to the fact that our last report included developments using Unity with the Vuforia developer kit [12]. Initially, it was of interest to research the possibility to create the tracking and pose estimation program from scratch, rather than using the popular Vuforia developer kit. This was desired to have greater flexibility when developing the AR system as well as gaining full control of all elements of the system. However, the evaluated tool setups described above all represent different difficulties, time sinks or the need for purchasing assets. All of these potential problems could be solved by using Unity with Vuforia, with the negative side being the loss of full developer control of the computer vision part of the system. Because the scope of this report is aimed towards AR and not computer vision (see section 1.3), the trade-off of letting the Vuforia algorithms handle the computer vision part to free up time to develop a more comprehensive AR application is assessed as a reasonable one.

The combination of running Unity with the Vuforia developer kit is a powerful tool when adding augmented information to a scene is desired. There are though, several engines and API's capable of rendering augmented information to a scene or a video stream like the Unreal engine, OpenGL and Vulcan. The reason for deciding to develop the AR system using Unity is solely based on the usability of the engine, the community supporting developers, and the convenient integration of the Vuforia AR developer kit. This setup solves many of the problems faced in other proposed and tested tool setups, however, it has its own limitations which require clever workarounds to fully manage to develop a complete AR system.

3.3 Developing an AR operator interface

When developing a system that aims to give subsea ROV operators increased depth perception when looking at a 2D monitor, the system running in the background has to understand the environment captured by the camera. In addition to this, the 3D information calculated by the system has to be presented to the operator in the most convenient manner possible to ensure great quality and usability of the system. Developing a system fully working on a subsea ROV underwater would be the ideal scenario for this report. However, this goes beyond the scope of this report (section 1.3). The contributed system described in this section is intended to function in air, without the visual disturbances of being underwater. The main goal of this system is to provide the user with real-time depth information and to support them when decisions have to be made. This should be provided in an intuitive and useful way. In the following section the tracking of the AR system is first described, then comes the breakdown of the designing and implementation process of the AR operator interface (the AR system).

3.3.1 Marker tracking and selection

In the developed system, Vuforia handles the tracking and pose estimation of markers discovered by a camera. The prior knowledge gained from developing a tracking and pose estimation program in OpenCV is very helpful to understand what Vuforia rely on to track markers in Unity. The same robust tracking, or maybe even greater, is achieved by Vuforia. This is done by loading the desired image target, which represents the ArUco markers in the OpenCV program, into their database with dimensional information about the real physical size of the image printed as a marker[87]. Then the algorithms in the Vuforia engine is creating a Unity compatible dataset asset that can be integrated into Unity to track this image target in a live video feed. Since Vuforia is not open source there is no way to know for sure how they do their tracking, but it is reasonable to think that they do some kind of feature detection of the image loaded into their database, like described above in section 3.1. While the scene is running in Unity, the Vuforia engine is probably constantly looking to match these features loaded in the database with features coming from the video feed to recognize and track the image target in the real world[88, 89]. With this in mind, image targets, or from now on markers, should be detailed with much texture and follow the recommendation for selecting great markers from section 2.1. This is desired so there can be a lot of features to extracted from each marker. This is ideal to help optimize the tracking of the markers in the scene[68].

After prior experience using ArUco markers, the initial thought was to use these markers with Unity as well. Nevertheless, these markers did not offer enough features for Vuforia to achieve optimal tracking. Therefore, another option using Vumarks[90], which is Vuforia's own markers, is tested. These markers are very customizable and have to be created individually one by one. This is evaluated to be a time demanding task with low reward, the focus is therefore shifted towards the use of QR-codes which can be easily generated online. While generating QR-codes and testing them, the Vuforia database is rating the tracking capabilities of these codes to 5 out of 5 stars, which according to Vuforia[68] means that these markers have a sufficient amount of features to make them optimal markers for AR development. Thus QR-codes is the used markers as they manage to utilize the full tracking capabilities of Vuforia at the same time as they are efficiently generated with the help of online QR-code generators[91].

3.3.2 Designing and implementing the AR operator interface

When designing and implementing the AR operator interface, we sat out to realize the concept proposed in our earlier work found in [12] and illustrated in Fig. 3.4. With some modification, improvements, and limitations (see scope 1.3) the vision from this previous work was implemented. Because we had no usability evaluations to work with when developing the user interface we had to anticipate and research what the operators might need and where to place it on the screen[92]. It was therefore decided to implement three different depth information AR-tools in the AR system. With three different depth tools, it is desired to learn what kind of tool works best during operations and discover positive and negatives with each of these tools. When designing this contribution the main drivers is to keep it as simple as possible, make sure every element has a purpose, and to strategically use color and texture to improve the user experience. In the spirit of this a general user interface (UI), which consist of a 2D border that provides information along the edges of the screen, was also added to the system.

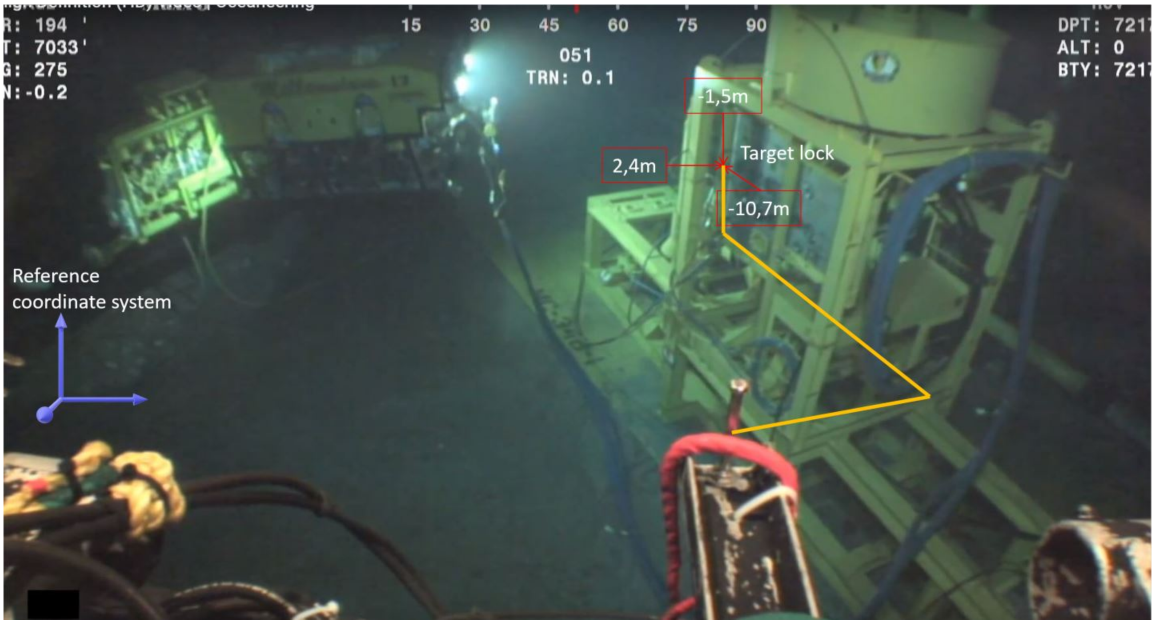


Figure 3.4: Illustration of the proposed ROV operator interface from our previous work [12].

3.3.2.1 Implementing AR with Unity

Since Unity is most know for being a game development engine, scenes are normally artificially made. With the help of the Vuforia developer kit, we get an asset named ARCamera which accesses the camera plugged into the computer. With this asset, the scene is no longer artificial, it is the area of the world captured by the camera. This asset is what enables developers to create AR applications. Moving on, 2D objects, basic 3D geometry or more advanced 3D models can be placed, moved about and scaled in the scene. These objects functions as the AR information in the scene.

For this implementation, the AR objects are built from 2D objects, basic 3D geometry, and some more advanced 3D models, which are created in the 3D CAD program, Creo Parametric 4 [93]. To make the scene and 3D objects dynamic, it is possible to attach self-developed C# scripts to the objects in the scene. This is roughly the basics of how AR is implemented in Unity with Vuforia. From this, it should be noted that the complexity of implementing AR objects into the scene is dependent on the complexity of the desired functionality of the AR objects. An example of this is the ease of just adding a static AR object on top of a marker like in Fig. 3.2. This is not very difficult when you have gotten familiar with Unity, Vuforia and the processes of implementing AR.

3.3.2.2 Depth AR tools

When the markers are detected in the scene, the position of the camera capturing the scene is estimated relative to the markers. In addition, the marker will have some position within the scene when it is detected by the camera. With the help of a C# script, the actual position between the marker and camera is estimated. That is, theoretically, the position values which yields zero in x, y and z-direction when the camera is exactly at the center of the marker. This is fairly simple to do with this software setup, and listing 3.1 shows how this is done in the C# script. It is also worth mentioning that the y-coordinate is the depth coordinate, x is right/left and z is up/down in the Unity scene.

```
void Update()
   {
       //// Collect the individual positions of marker and camera
       ToolPoseX = ToolMarker.transform.position.x;
       ToolPoseY = ToolMarker.transform.position.y;
       ToolPoseZ = ToolMarker.transform.position.z;

       MarkerPoseX = marker.transform.position.x;
       MarkerPoseY = marker.transform.position.y;
       MarkerPoseZ = marker.transform.position.z;

       //// Calculate the distance between marker and camera
       poseX = -(MarkerPoseX - ToolPoseX);
       poseY = -(MarkerPoseY - ToolPoseY);
       poseZ = -(MarkerPoseZ - ToolPoseZ);

       //// Add the calculated relative position in a 3D vector
       Vector3 Pose = new Vector3(poseX, poseY, poseZ);
```

Listing 3.1: Calculation of position between marker and camera in Unity.

This calculated information is essential to have when you want to give operators increased depth perception. To utilize and present this information to the operators, three AR depth tools were designed and developed. This estimated distance between marker and camera is used in all three of these depth AR tools. What differs them from each other is simply the way the depth information is presented to the operator. In the next paragraphs, the thought behind and the implementation process of these three different depth tools will be presented and described. Is should be noted that the movement of the camera capturing the video feed is referred to as the movement of a vehicle. This is to closer relate the implementation to an ROV moving about in some arbitrary scene with some markers in it.

Path

The proposed path design in Fig. 3.4 (the yellow lines) from [12] is attempted to be implemented in the AR system. The intention of this tool is to show operators how they can move their vehicle to reach their target through dynamic lines. In Unity, this path was created using the built-in line renderer function, which uses the pose estimation to draw the path lines in all three dimensions. However, because the Unity scene is updated with every frame at about 30fps, the lines are updated very frequently. Because of this, they were heavily exposed to noise coming from the pose estimation. A solution to dampen this noise, which suddenly would rapidly increase and then degrees the length of the path for a very short time, was to implement an averaging filter. With this filter, the distance estimates fed to the line renderer is the average of the last four estimates[94]. This makes for a more stable path with less sudden changes in the drawn lines. The averaging filter is used in all of the depth tools and how it is developed with C# code is presented in listing 3.2.

```
void Update()
   {
        ////Pose estimation filter

       // Add pose estimates to pre-defined lists
       PoseEstX.Add(poseX);
       PoseEstY.Add(poseY);
       PoseEstZ.Add(poseZ);

       // Introduce a count variable
       i = PoseEstX.Count;


       // Calculate the average of the four last pose estimates and
          assign it to a new variable.
       foreach (float value in PoseEstX)
       {
           EstX = (PoseEstX[i - 4] + PoseEstX[i - 3] + PoseEstX[i -
              2] + PoseEstX[i - 1]) / 4;
           EstY = (PoseEstY[i - 4] + PoseEstY[i - 3] + PoseEstY[i -
              2] + PoseEstY[i - 1]) / 4;
           EstZ = (PoseEstZ[i - 4] + PoseEstZ[i - 3] + PoseEstZ[i -
              2] + PoseEstZ[i - 1]) / 4;
       }

        // Assign the filtered estimates to a new 3D vector
        Vector3 FilteredPose = new Vector3(EstX, EstY, EstZ);
```

Listing 3.2: Pose estimation filter to reduce noise in estimation.

Different from the proposed design, the path is equipped with changing colors based on how close the operated vehicle is to the target. It has a gradient color scheme going from red to blue. Red indicates that the operator is far from the target and when it is moving towards blue the operator is getting closer to it. This change in color is possible through animating the developed path in Unity, with a function called lerping [95]. In addition was a criterion coded to turn of the path after the target was reached, so that it did not take space in the scene when it was no longer needed. This is included together with the rest of the functionality of the path in the C# scripts presented in appendix C. Probably the most difficult aspect of implementing this path was to make sure the lines were integrated into the observation of the physical world in a good manner and to make sure the path actually was drawn from the origin to the target.

Coordinate system

Another implemented depth tool is a regular three-dimensional Cartesian coordinate system. The coordinate system is a standard Windows 3D paint model that is converted to a 2D sprite element and then implemented in the scene within Unity. This coordinate system shows the operator the positive defined directions in the scene. Next to each arrow in the coordinate frame is a 3D text box containing a number which indicates how far the operated vehicle is from the target. This is illustrated in Fig. 3.5, where "X: 6,2" tells the operator that the vehicle is 6.2 centimeters to the right of the target. If the vehicle is maneuvered 6,2 centimeters to the left, the X value would go to zero. In the figure, one can also see the use of colors to distinguish the three directions from each other and the addition of a background to ensure high contrast.

The motivation behind implementing the numbered distances is the thought that some people might prefer to relate distance to numbers, numbers which in this case is dimensioned in centimeters.

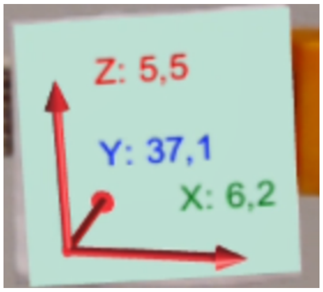


Figure 3.5: The coordinate system implemented in the AR system as one of the depth tools.

Depth bar

The last implemented component giving the operator depth information is a depth bar implemented to the very left in the operator interface. This component is illustrated in Fig. 3.6. It is based on the same gradient color scheme as the path uses, where red is still far from the target and when it turns blue the operated vehicle is still getting closer. The green slider bar is a 3D model created with Creo Parametric [93] and it is what illustrates the distance between the vehicle and the marker. It movies up and down along the depth bar depending on this distance. In Unity, the depth bar is implemented by customizing the functionality of the built-in slider component with a C# script, which can be found in appendix C. The visuals are also modified with a self-created gradient color in Unity. This AR depth bar is only providing depth of field information, i.e., information in y-direction. Because of this, it does not say anything about the whereabouts in z- and x-direction. This is intended as the thought is to test how useful a graphical presentation of the depth information is compared to the other solutions.

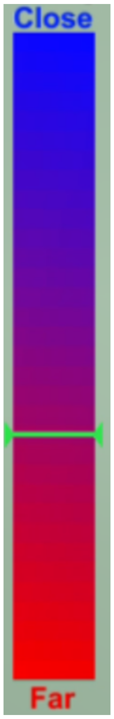


Figure 3.6: The depth bar.

3.3.2.3 General user interface (UI)

To give the operator more information during operations a border with information presented in 2D is added to the contributed AR system. This border, see Fig. 3.7, serves the purpose to keep operational data close to the line of sight of the operators so that the focus could be on the monitor with the video feed as much of the time as possible. Therefore different information components are added, these include time and date, a list of all task to be done, oil and gas field information, water depth, current tool, and relevant buttons which serves a purpose in different use cases. The depth bar is also integrated as a part of this border. In Unity, a canvas is placed on top of the scene captured from the camera to build the border. By introducing this canvas, 2D objects can easily be dragged and dropped around in the scene and the dynamic objects in the border such as operation timer, water depth, time and date, buttons and the tool indicator are all attached to different C# scripts found in appendix C.

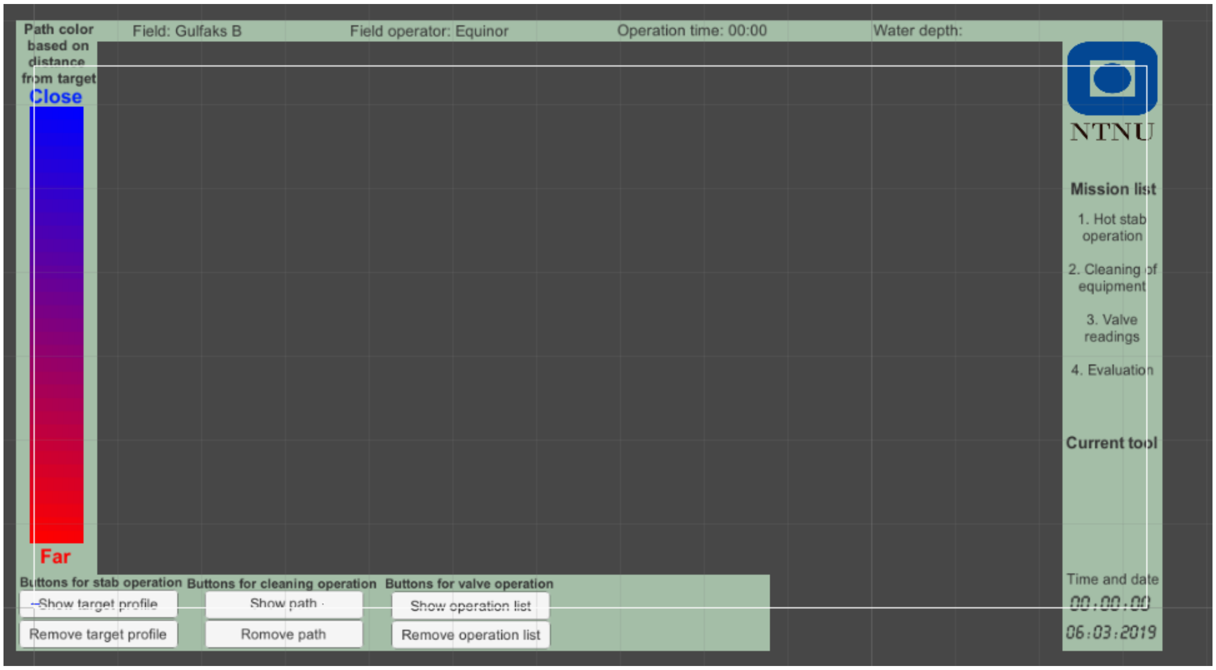


Figure 3.7: Illustration of the general UI designed and presented inside Unity.

Chapter 4

The design, arrangement, and conduction of the experiment

We have in the previous chapter seen how the AR system or AR operator interface has been developed and implementing containing different depth tools and a general UI. It is now desired to test the performance of the system and collect user feedback about the usability of the system. To do this, different operations are designed and the AR system gets additional features to better support the users through these operations. The goal of this chapter is to describe how the experiment was designed, arranged and conducted. This is done by first looking at a description, then the participants and the setup, before the actual operations and evaluation within the experiment is presented. It should be noted that the users of the developed system are referred to as test participants or simply participants within this chapter, and in the next when referring the system during the experiment.

4.1 Description and purpose of the experiment

The experiment will be executed by eight test participants. All eight participants are fellow students. The experiment with the evaluation will take approximately 60 minutes. In the experiment, each participant is supposed to operate a tool and perform three different operations with this tool. The only vision of the operation available to the operator is a video feed presented on a 2D monitor. First out the participants go through all the operations with only the pure video feed from the camera (the non-AR system), and in the second run, the AR system which aims to provide depth perception and decision support is used and followed by an evaluation.

The purpose of the experiment is to evaluate the value of having 3D information presented on a 2D monitor while operating some equipment in a 3D environment. The experiment also serves the purpose of providing user feedback about the developed system to discover positives and negatives. In addition, it aims to investigate the potential of supporting operators in making decisions while using the system. It is also the purpose of this in air experiment to in a simplified way, simulate the hassle of operating a subsea ROV with only visual information coming from a camera sensor. This purpose emerges from the desire to investigate the research question presented in section 1.2.

Figure 4.1: The complete experiment setup in the experiment room at the department.

4.2 The volunteering participants

The whole experiment including all the operations and evaluation is aimed to take about 60 minutes and is carried out with eighth participants. The number of participants was based on Jakob Nielsen's theory that elaborate usability tests are a waste of resources, and that the best results come from testing with no more than 5 users while running as many small tests as possible [96]. From earlier research, he and Landauer in [97] found that the numbers of usability problems found in a usability test with n users follow Eq. 4.1.

$$Usability\ Problems\ Found = N(1 - (1 - L)^n) \tag{4.1}$$

In this equation N represents the total number of usability problems in the design, L is the proportion of usability problems and n is the numbers of test users. Nielsen has averaged the L value across a large number of projects and found the average value to be $L = 31\%$, this gives the following results presented in Fig. 4.2. From this graph, one can see that about 85% of the usability problems are discovered with only five test participants. This method recommends performing usability evaluations with fewer participants more frequently. In this experiment, however, the time limited us to only perform one usability evaluation. Therefore, eight participants were used so that about 95% of the potential usability problems could potentially be discovered during the experiment.

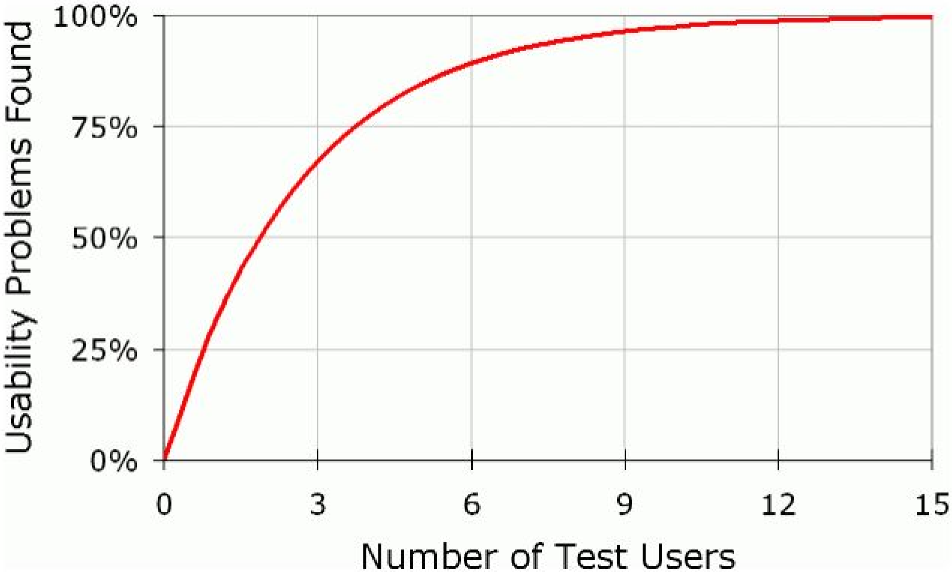


Figure 4.2: Jakob Nielsen and Thomas Landauer's model of usability problems found as a function of number of test users [97]. This is essentially Eq. 4.1 plotted with $L = 31\%$, source: [96].

Of these eighth participants, two of them are females and six are male. They are all students at NTNU Gløshaugen enrolled in four different engineering disciplines. Five out of eight participants are in their fifth year, two are in their fourth year, and the last one is in the third year. When the participants are asked if they have any prior experience with operating a drone from a 2D display, only two replies some, while the rest have no such prior experience. They were also asked to rate how familiar they were with AR prior to the experiment on a scale from 1 to 10. Their answers are presented in Fig. 4.3 and are together with the rest of the collected evaluation responses, collected anonymously.

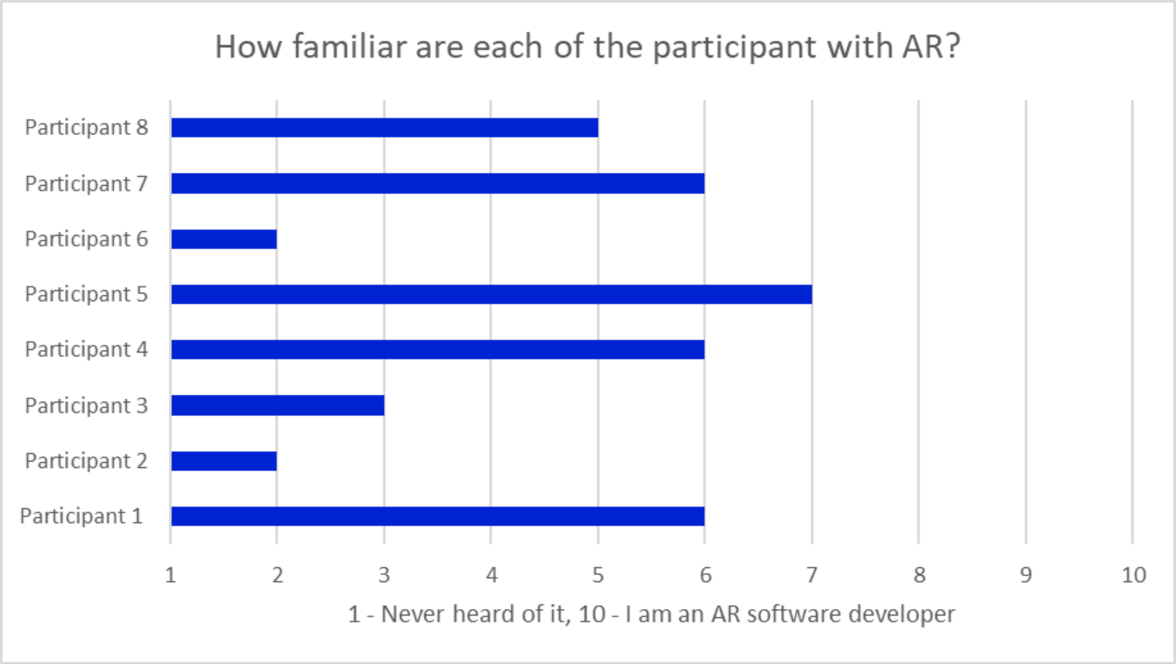


Figure 4.3: Mapping of each participants prior familiarity with AR.

When the participants enter the room where the experiment is executed, they are handed a description of the experiment and the operations, the handout is found in appendix A. After the participants have read the handout they have the opportunity to ask any questions if something is unclear. Next on, they have some time to familiarize themselves with the equipment used in the experiment (i.e., the tool introduced in the next section), before the mission with three different operations starts. From the second the participants enter the room and are handed the description, they have a maximum of 10 minutes to read, ask questions and get familiar with the tool before the operations start.

4.3 Physical setup of the experiment

To carry out the experiment, a test rig is needed. This was designed and developed with the intention to let test participants complete a set of operations to test how the contributed AR system performs in the hands of others. The complete setup is illustrated in Fig.4.1 and consist of a self-created tool, a camera, the AR scenes and more. This setup was designed with the intention to execute some physical operations within this experiment and these setup components will be described in this section.

4.3.1 The tool

A manually operated tool was developed so the test participants had some equipment to operate to complete the experiment. The core of this developed tool is a 2-meter long wooden beam. At the end of this beam is an attachable horizontal extension. This was developed to simulate the fact that subsea ROVs have a wide range of different tools they can use during different operations [98]. Another intention is the idea that a horizontal extension could rely even more on precision and depth perception during different operations. With this wooden beam and the extension, the tool has two different modes. One basic vertical mode with a regular I-profile, and one more advanced horizontal mode with an L-profile. These modes are illustrated in Fig.4.4 and Fig.4.5 respectively.

Figure 4.4: The tool in I-profile mode with a camera attached.

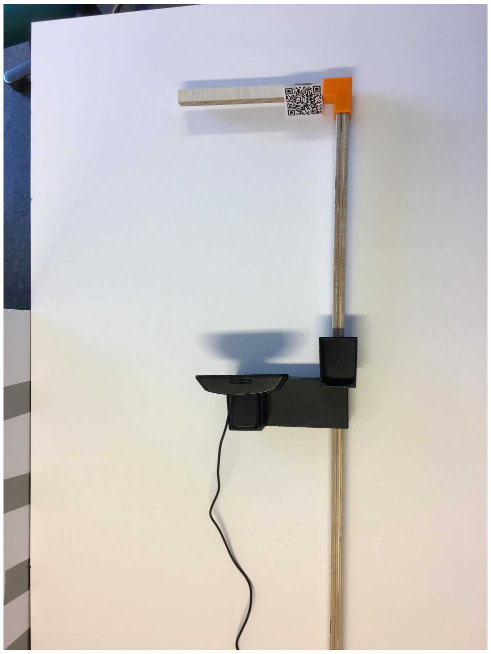


Figure 4.5: The tool in L-profile mode with a camera and a marker attached.

To give the operator the opportunity to maneuver the tool, an in house designed and developed mount containing a ball-joint was 3D printed and illustrated in Fig. 4.7 and B.3. This mount gives the operator 5 DOF to maneuver the tool about with, i.e., forward/backward, right/left, up/down, pitch, and yaw movement. The maximum potential of 6 DOF is illustrated in Fig. 4.6 and roll are the one constrained DOF in this setup. These 5 DOF are however somewhat constrained by the mount, the wall (seen in Fig. 4.1) and the length of the horizontal beam. Therefore, the operations have to be designed within these constraints. To see the maneuverability of the developed tool, click on Fig. 4.8 to start a video[1]. By keeping one hand directly on the rear end of the wooden beam and the other hand on the handle on top of the tool mount also seen in Fig. 4.7, the operator is in the best position to directly control the 5 DOF. Due to the length of the beam and the equipment placed on it, there is some weight to the tool and the participants were, therefore, given time to get a feeling for the force needed to control it.

---

[1] The video is only possible to run with the latest version of Acrobat Reader [99].

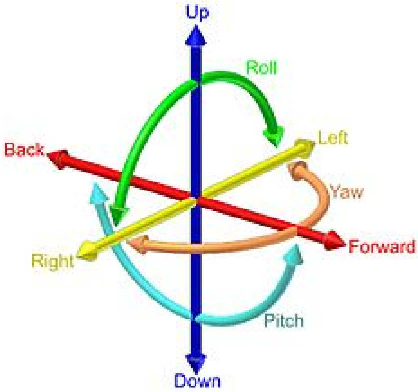


Figure 4.6: Illustration of the potential degrees of freedom.

Figure 4.7: The ball joint mount used to control and operate the tool.

Figure 4.8: If clicked plays a illustrating the maneuverability of the developed tool.

4.3.2 The camera

With the purpose of this experiment being based around a participant operating some tool by looking at a video stream presented on a 2D monitor, naturally, a camera has to be a part of the experiment setup. The camera used is a Logitech webcam (model C925e) which is eligible to record video with a resolution of 1920x1080 (1080p) at a frame rate of 30 fps [100]. This camera is conveniently placed about 40 centimeters from the end of the tool in two different custom designed and 3D printed camera mounts. This opens the door for potentially having two different camera positions. These two camera placements can be seen in Fig. 4.4 and 4.5 and models of the mounts can be found in appendix B. Because the tool is capable of operating with two different modes (i.e., I- and L-profile), which shifts the position of the tooltip, it is handy to have the option with two different camera positions as well. Therefore, the camera mount used while the tool is in the L-profile setting has a horizontal extension out from the main beam, shifting the field of view towards the tip of the horizontal beam extension. The camera mount used during I-profile mode, on the other hand, enables the camera to simply be mounted right on top of the vertical beam. Due to this flexibility, it is always possible to keep the tooltip in field of view even if the tool mode is changed. The disassembled tool, camera mounts, and the camera are shown in Fig. 4.9.

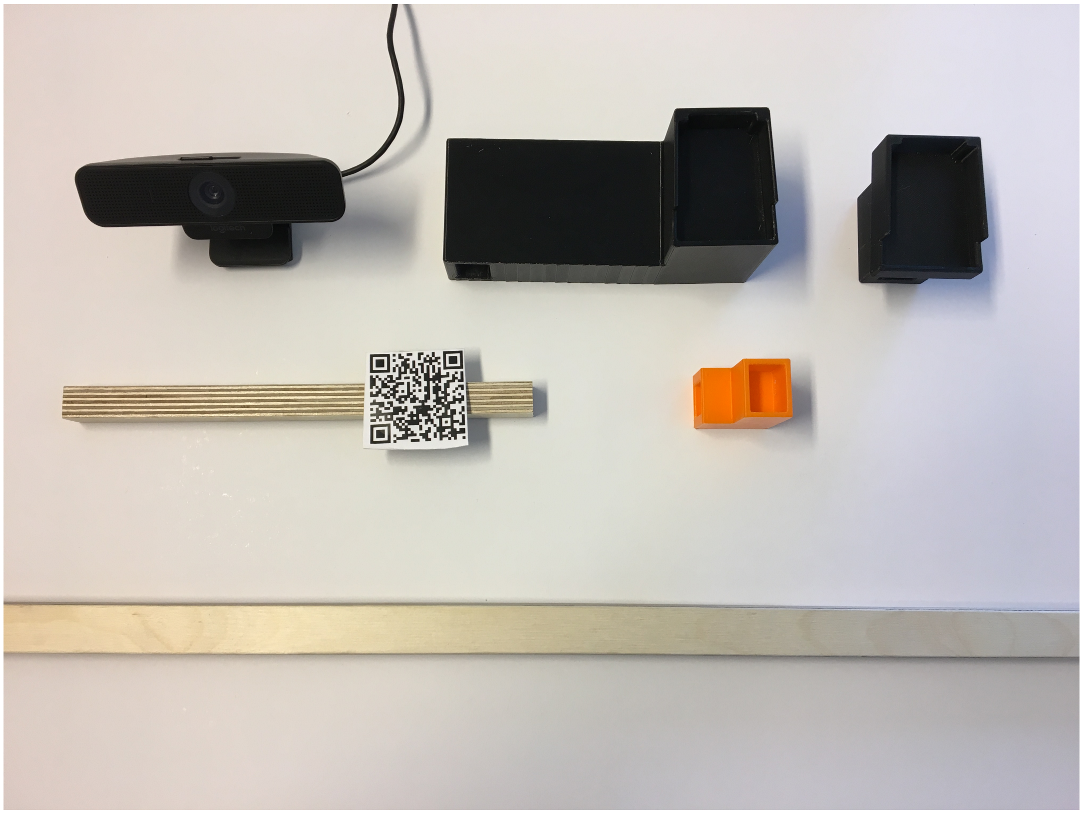


Figure 4.9: The disassembled tool with from the bottom the wooden stick, the horizontal extension and joint, the two different camera mounts and the camera itself.

4.3.3 The complete setup

With a maneuverable tool and a camera sensor to record its movements, the fundamental parts of the setup are in place. However, other components are needed to fully complete the experiment setup. In order to follow the purpose of this experiment from section 4.1, walls have to be installed to block the participant's line of sight. The temporary installed wall is placed on top of a pair of writing tables and it has a cutout where the tool mount is installed. The cutout has to be as small as possible to reduce the ability for the test participant to look through the cutout rather than on the 2D monitor. However, by reducing the cutout section the constraint on the movement of the tool is increased. Therefore, the sweet spot between mobility and visibility through the wall is found and presented in Fig. 4.10. This figure illustrates the size of the used cutout which does a great job of blocking visibility through the wall, while still ensuring full maneuverability within the operation scenes.

Figure 4.10: Illustration of the visibility out of the wall and constraint on the movability of the tool.

The wall divides the test setup into two segments. The front side which is the only side available to the participants and the backside where the operations are physically executed, see Fig. 4.11. At the front side, each of the participants is met with a 12 inch Surface Pro screen, the tool control station (rear end of the tool), and a table with additional operational data needed to complete one of the operations. Behind the wall, one can find the main part of the tool, a 3D printed docking to dock the tool in-between operations, and the different operation scenes. These will be introduced in the next section.

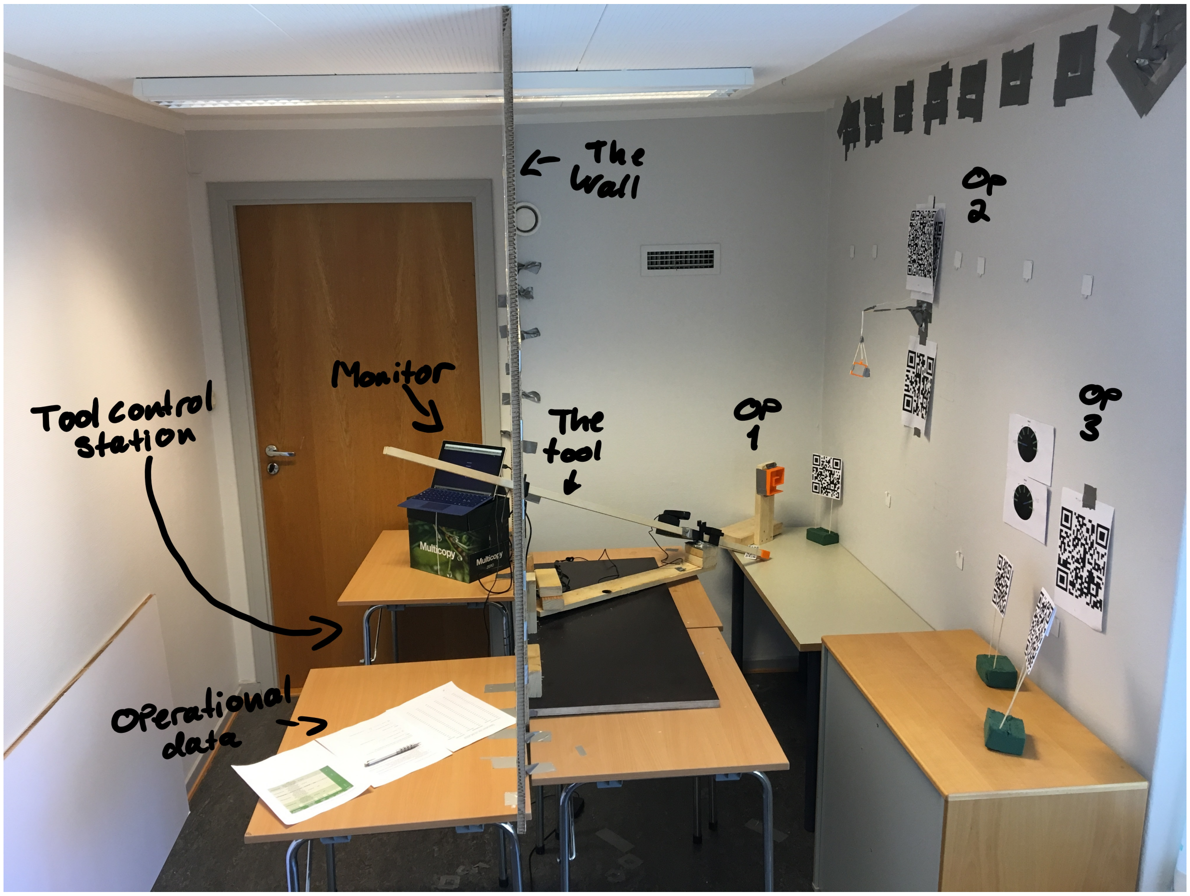


Figure 4.11: The complete setup in the room provided by the department, with indicators.

4.4 The operations

In order to test the depth perception and decision support capabilities of the developed AR operator interface (section 3.3), three different operations are designed. Based on these goals, operations 1 and 2 are mainly focused on testing the participant's perception of depth in the scene through two different stab operations. Meanwhile, operation 3 is the one operation really testing the decision support capabilities, through a valve readings operation. It should be noted, however, that decisions support also plays a role in operation 1 and 2. In the AR system, AR components have been developed to provide decision support capabilities through operation specific AR information in every operation. These operations will be further explained in this section and the participants will execute all of these three operations first without the AR system (non-AR system) and then with the AR system.

Operation 1

The first operation is supposed to resemble a subsea ROV stab operation where the operator has to move the tool through a certain profile to access the stabbing point [101, 102]. It is therefore not straight forward to access the point of which the tool is supposed to be stabbed. A subsea stab operation was taken as inspiration when developing this operation so that the experiment could stay relevant in terms of subsea IMR operations, even though the test is conducted above sea level, with a wooden beam. The stabbing profile in this operation was, however, designed to be a bit more difficult than a straight forward stab, this is to test the support capabilities of the implemented AR.

With the intention above in mind, this operation consists of a 3D CADed and 3D printed maze with a square hole in the middle. The maze together with a marker is placed alongside each other in the operation scene, illustrated in Fig. 4.12. In this produced scene, the goal is to stab the tool through the center of the orange maze. To reach this center, the tool has to be maneuvered through the maze, which has a profile layout not visible for the test participant due to the camera angle. This is intentional. The operation is only successfully executed if the participant enters the maze all the way back, on the lowest section of the maze, i.e., the entrance which is marked with a circle in Fig. 4.12. In addition to the maze, there is as mentioned a marker which supports the AR scene with tracking and 3D information. This is the marker used by the pose estimation script highlighted in listing 3.1. The marker also displays the maze-profile as AR information, as seen in Fig. 4.13, to help the participant understand how the tool must be maneuvered to reach the hole. The maze profile is, however, completely unknown to the participants during the non-AR part. They, therefore, have to feel the walls of the maze and navigate to the center solely based on this sensing information. It is thus interesting in this operation to collect information about how the participants act when the maze layout is unknown compared to know with AR. Another interesting aspect is to evaluate if the AR system manages to help the participant find the entrance of the maze through the implemented depth tools (section 3.3.2.2) and operational based AR information. To get a better understanding of how operation 1 is executed, click Fig. 4.14 to start a video of one of the test participants run through the operation without the AR system.

Figure 4.12: Operation 1 experiment scene, with the orange maze where a drawn circle marks the entrance.

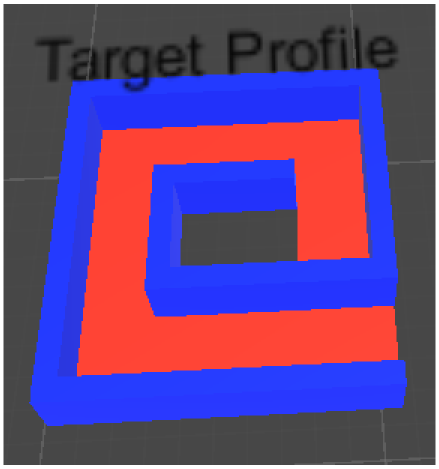


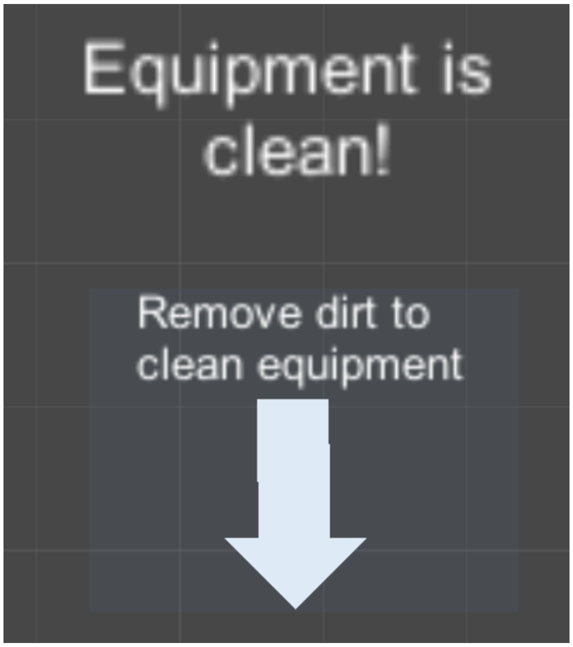
Figure 4.13: The operation specific AR component in operation 1, which provides the maze profile.

Figure 4.14: If clicked plays a video of operation 1 without AR.

Operation 2
Cleaning operations are a big part of the day to day tasks performed in ROV operations [36]. To simulate something similar in this experiment, a method to take advantage of the depth of field information in the AR system is desired. So, an operation where the objective is to remove "dirt" from a marker was designed. The dirt, which in this case was an A4 sheet of paper with a marker on it, can be removed by hooking the tool into a quadratic hook. This is another type of stabbing operation, but different from operation 1, high level of accuracy and finesse is necessary to properly insert the tool. If the participant is not taking care when inserting the tool, the hook might be moved and fall off the tiny supporting structure holding it in place, see Fig. 4.15. This makes for a potentially more demanding operation, in which the depth of field information could be beneficial to reduce potential stress and frustration. Therefore, as in operation 1, there is a marker behind the quadratic hook used for tracking and 3D information.
The dirt is removed by inserting the tool into the hook and then by pulling the tool down to clear the “dirt”. To complete the operation the participant must maneuver back to the cleaned marker to check if it is completely clean and then confirm to the practitioner that it actually is cleaned.

During the first run, the test participant is supposed to hook the tool and confirm that the marker is cleaned without any support from AR. In the second, the participant has all the depth tools, presented in section 3.3.2.2, available together with an indicator showing where the hook is placed and where to stab it from. In addition, a decision support message telling that the marker is cleaned appears if the AR system manages to recognize the marker by it satisfying the Vuforia tracking criteria (section 3.1). This operation specific AR information is presented in Fig. 4.16. Like for operation 1, it is of interest to see if the AR depth tools are to any help for the participants when searching for the required position to insert the tool through the hook. Another aspect is to observe how confident the participants are when claiming whether or not the marker is cleaned and if the supportive confirmation from the AR system is to any help. Same as for operation one, if Fig. 4.17 is clicked a video of a non-AR run through the operation will be presented.

Figure 4.15: The operation 2 scene, with the quadratic hook placed on fragile wall mount.



Equipment is clean!

Remove dirt to clean equipment

Figure 4.16: The AR specific component for operation 2, indicating where the target is and shows a "cleaned" message when the tool is cleaned.

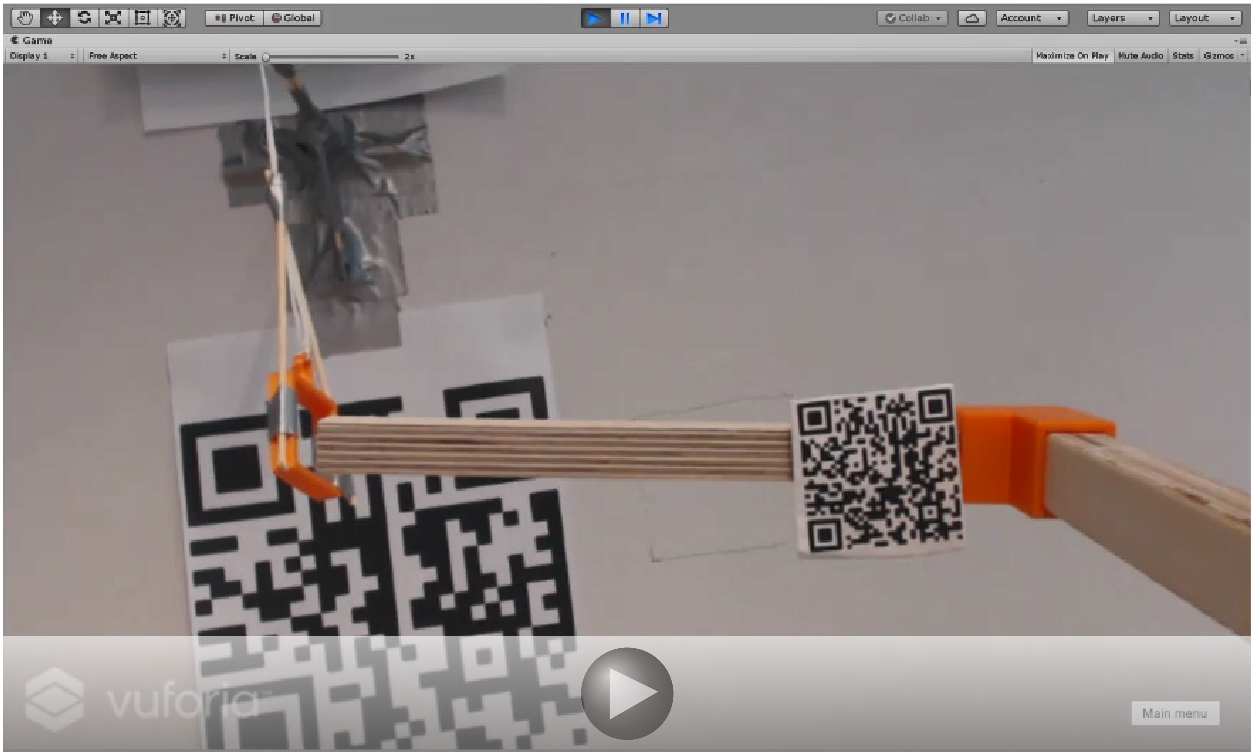


Figure 4.17: If clicked plays a video of operation 2 without AR.

Operation 3

During subsea ROV missions, the ROVs are passing several valves while moving from operation to operation. This lets for potential "drive-by" manual valve readings by the operators. So, in this valve reading operation, the participant is supposed to manually check the choke valve position on some subsea equipment. Dummy valves were used in this operation and these complete the scene together with some markers used for AR information illustrated in Fig. 4.18. In the non-AR system, the participant must move the tool and camera close to the dummy valves to read their status, then dock the tool and go through an operation list, an operation database, and a valve operation guidebook which is handed out in paper format. These handouts can be found in appendix A and they are placed next to the participants during the operation (see Fig. 4.20). These handouts resemble operational information available to real subsea ROV operators through their field information databases.

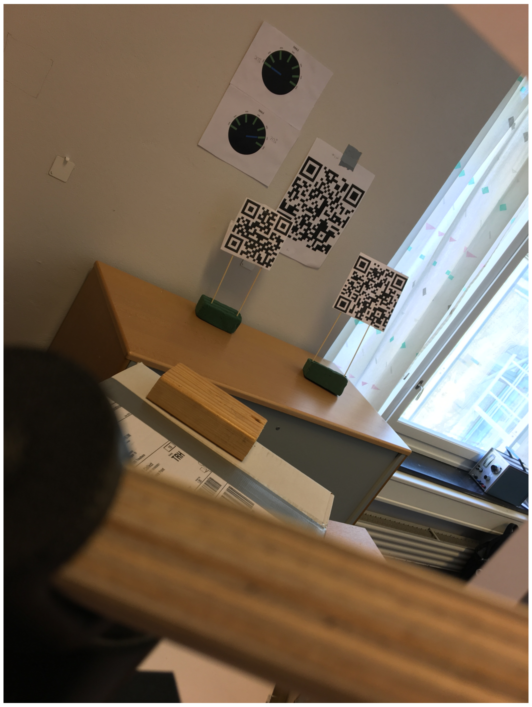
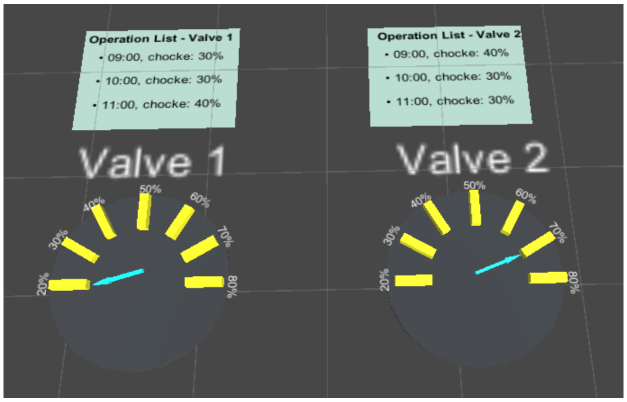


Figure 4.18: Operation 3 experiment scene, with the valves mounted on the wall used in the non-AR part, the standing markers are valves in the AR system and the wall mounted marker is a reference for the dialog boxes.



Operation List - Valve 1

- 09:00, chocke: 30%
- 10:00, chocke: 30%
- 11:00, chocke: 40%

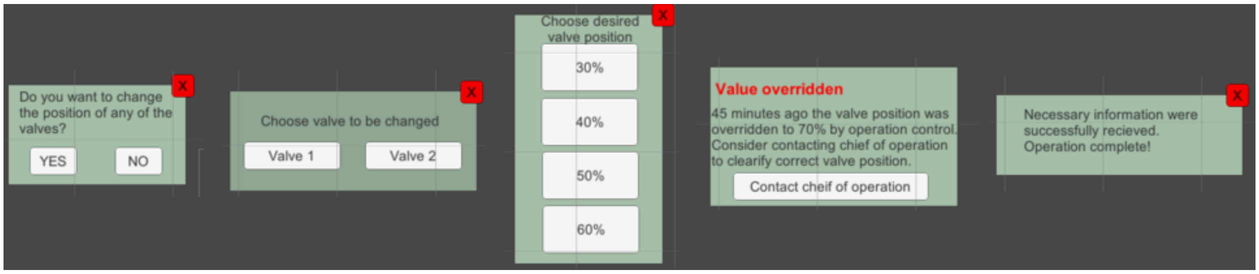
Operation List - Valve 2

- 09:00, chocke: 40%
- 10:00, chocke: 30%
- 11:00, chocke: 30%

Figure 4.19: The AR specific component for operation 3, which highlights the view of the valves and provides an live operation list in line of sight.

Figure 4.20: The three handouts are placed on the right side of the participant which stands in-between the two tables during operation.

In the operation, none of the valves are in the correct position, and it is up to the participants to decide what to do with the valve position of each of the chokes with the handed information. The operation is completed when a decision is written down and presented to the practitioner. When the same operation is performed with the AR system, the information handed out in paper format is integrated into the system. This way it should be easy for the participant to understand what to do with the valve positions. By following dialog boxes presented in the AR operator interface, the participant should get enough support from the system to make the right decisions. The operation with AR is completed when the participants reach the last dialog box of the button sequences displayed in Fig. 4.21. The inspection of the valves and docking of tool part of the non-AR operation is shown in the video in Fig. 4.22.



Do you want to change the position of any of the valves?

Choose valve to be changed

Choose desired valve position

Value overridden

45 minutes ago the valve position was overridden to 70% by operation control. Consider contacting chief of operation to clearify correct valve position.

Necessary information were successfully recieved. Operation complete!

Figure 4.21: The button sequence developed to execute operation 3 with AR. The sequence goes from left to right.

Figure 4.22: If clicked plays a video of operation 3 without AR.

4.5 Assumptions and limitations of the experiment

An ultimate goal would be to implement a functional AR operator interface with usable depth perception and decision support capabilities directly in the interface between subsea ROVs and their operators. This is tough, out of the scope of this report (section 1.3). The experiment will, therefore, be conducted in air, above sea level, to eliminate the disturbances from water turbidity, loss of light, and life underwater [12]. In addition, a limitation is the use of marker-based AR, which makes the process of implementing AR easier, but introduces the need for several markers in the test scene and limitations in the distance before the camera manages to detect the marker. When using a marker-based solution the test also has to be pre-calibrated by taking physical measurements and then adding these to the pose estimation C# script (listing 3.1). There are also limitations in movement, whereas the tool can only be maneuvered within the experiment setup area, thus the operation scenes have to be within this area. And in this scene, everything is static other than the maneuvered tool with the mounted camera on it.

An assumption is that the developed tool should represent the right manipulator or a certain equipped tool on a work class ROV, see section 2.2.1 for more information about these. They are of course different in complexity and advancedness, however, they both are operated by an operator (one with electrical controls and one with physical controls) and both have a camera providing live imaging from its environment. Another one is that each of the operations is assumed to simulate different familiar subsea ROV operations, to make the test operations somewhat relevant to the subsea IMR aspect.

Even though there are a set of assumptions and limitations within this experiment, these are within the lines of the research question because this question focuses on the usability of the integration of certain mentioned capabilities in the AR operator interface. The subsea IMR part is also included to keep the experiment close related to the goals of this report (section 1.2).

4.6 Description of the evaluation process in the experiment

To collect feedback from the participants after they have used the experiment setup and AR system, the experiment is ended with an evaluation. The evaluation will be done by surveying both the participants and the practitioner. The survey answered by the participants highlights three different evaluation methods. A general subjective evaluation, usability evaluation, and a system usability scale (SUS) evaluation. Meanwhile, the practitioner survey is there to log the performance of each participant and interesting expressions from the participants during the execution of the experiment. These evaluation methods will be described in this section and in the next chapter, the evaluation results will be presented and discussed.

4.6.1 Evaluation survey

After each test participant has completed the operations both with and without AR they have to do an evaluation of the tested system. This is done through a survey with over 90 questions and statements. With this survey, it is of interest to gather qualitative and quantitative results about the usability and the decision support aspect of the AR system. Through this survey, the system will be evaluated both on component level (i.e., the components will be evaluated up against each other) through a subjective evaluation and as a whole functioning system from a usability perspective. The usability evaluation of the whole system will be in two parts. One looking at just the qualities of the AR system as a standalone system and the other comparing the qualities of the AR system with the performance of the non-AR system. This comparison is done with questions asked on the form "this system performed better than this in that area". These questions were asked by alternating which system was mentioned first in the sentence in an attempt to eliminate or reduce any obvious bias towards the developed AR system.

The reason for evaluating usability and the other factors mentioned above is because these act as different pillars which together plays significant importance when trying to create a great user experience [103]. To integrating the usability goals (section 2.3.1) and evaluation of the components in the survey, several types of questions and statements are made with different types of answering methods. The main answering methods are liner scales ranging from 1 to 5, yes/no/not sure, and agree/disagree/not sure answers to statements. There are also some multiple choice questions and short answering questions with the intention to potentially collect a wider range of information from the participants. Each participant must go through the survey immediately after the operations are completed, in the same room as the experiment is conducted. The survey is completed on a computer through Google forms [104], and the answering setup is shown in Fig. 4.23.

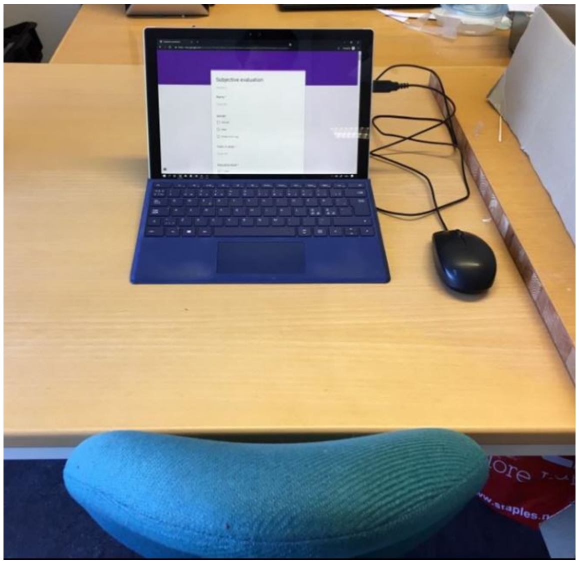


Figure 4.23: The survey is completed on computer in the same room as the operations are executed.

There is also a survey that the practitioner has to complete after each participant has been through the experiment. In this survey more objective factors such as the time used while completing each operation and the number of times, the participant stabs the horizontal tool top to the left in the air is logged. When operating only with 2D information, the stabbing kind of relates to when you are navigating in a known environment in pitch darkness and use your hands to try to feel where you are. The stabbing works in a similar way because one of your senses is taken from you. i.e the depth sense and you use the tool to try to feel where you are in the depth field. The amount of stabs is, therefore, logged because such stab mid-air could represent that the participants believe the tool is at the stabbing point before it really is. The stabbing point is in this case where they are supposed to enter the maze in operation 1 and when they actually are next to the to quadratic hook in operation 2. A video showing these counted stabs is presented by clicking on Fig. 4.24.
This survey also asks for subjective evaluation points such as distinctly expressed body language or verbally expressions during the execution of the operations. The questions used in the survey can be found in experiment evaluation file in the enclosure.

Figure 4.24: If clicked plays a video illustrating the stab in empty air.

4.6.2 Subjective evaluation

When evaluating the components of the developed AR operator interface (section 3.3) through a subjective evaluation, the components are separated into two different categories. That is one group containing all the AR depth tools (i.e., path, coordinate system, depth bar), and another containing the more operation specific AR information. The latter one includes the unique AR components for each of the three operations introduced in this chapter, plus the general UI which provides the general operational information. This is so that components that serve a more similar purpose easier can be evaluated up against each other.

The first category of AR information is evaluated based on seven different areas, these are listed below. For the second category focus is aimed towards the first five areas, excluding navigate because these components do not offer depth information. Color is also excluded because it is chosen arbitrarily to fit the experiment scene and theme of the layout.

- Helpfulness - Does the component help the participant execute the operation?
- Intuitiveness - Is the component easy to understand and intuitive to use in the operations?
- Necessity - Is the component necessary when completing the operation?
- Relevance - Is the component relevant in the context of these operations?
- Size and location - Is the component placed conveniently in the operator interface with size and location that does not block important visibility?
- Color - Is the color and contrast of the component suitable for the environment it is used in?
- Navigate - Does the component make it easier to navigate in a 3D environment, even though it is presented on a 2D screen?

The reason for choosing these focus points is that they are the main drivers used in the development of the system. Each of the represents areas seen as essential when creating components which adds value to the system. They are also chosen to help give indications on whether or not the component design manages to fulfill its potential.

4.6.3 Usability goals

In this section, the usability goals will be presented in the context of this experiment and they will represent the usability evaluation of the AR system. These usability goals is chosen based on the human - system interaction ISO 9241-210 standard [41], other literature within human-computer interaction [48, 43, 44, 42, 45], and the research question (section 1.2). Through the next paragraphs, the way each of these goals is focused on and how they are represented in the survey is described. To read more about the definitions of these goals go to section 2.3.1.

Effectiveness

The effectiveness looks at how precise and completely the participant is able to execute the operations [41]. For the objective evaluation, the number of stabs in the air will be an indication of the preciseness in operation 1 and 2, and for operation 3 the participant's ability to complete the operation is the measuring point.

For the subjective evaluation, the effectiveness questions will focus on the participant's ability to execute the operations with the tools at hand as well as the performance and robustness of the AR system.

Efficiency

Efficiency is all about doing the right thing in order to achieve a goal, i.e., reaching a goal without wasting resources [41, 50]. The objective evaluation will, therefore, look at the time used on the different operations both with and without the AR system. While questions about the participant's capability to do what's needed in order to execute the operation efficiently will have the main weight in the subjective evaluation.

Satisfaction

With satisfaction you want the operator to have the freedom of discomfort and positive attitudes towards the use of the system [41]. With this in mind, this goal will only be evaluated subjectively through questions about the participant's user experience with the AR interface and potential statements or notable body language from the participants during the experiment.

Workload

To get an impression of the mental, physical and temporal workload [43], the participant's ability to grasp all the information presented in the handout (appendix A) and execute the operations correctly is evaluated. In addition, the participants will be asked about the stress level of executing the operations and if the operations were less demanding (mentally and physically) to execute with or without the AR system. Workload will solely be evaluated subjectively.

Another method to measure the workload is through a NASA-TLX assessment [105], such an assessment provide an overall workload score based on a weighted average of ratings on six different subscales, these are mental, physical and temporal demands, own performance, effort, and frustration. It was, however, decided not to take advantage of this method as the survey was getting very long and some measures had to to be taken to reduce the length in respect of the participants and their own study time.

Situational awareness

SA looks at the participant's ability to collect and use information coming from the camera in order to execute the operations properly. Through the survey, participants are asked indirectly about the SA with the AR system compared to without. There are also more questions regarding the performance of the AR system to subjectively evaluate how the system affected the situation awareness [44].

Learnability
It is important that it is easy to get started with the developed AR system and that the participant can learn how to do a range of operations within a short period of time. With this in mind, the subjective evaluation seeks to find out if it is tedious to learn and understand the developed AR system and if it was easy for them to get familiar with the AR information [42].

Decision support
To evaluate the systems ability to help the participants in the process of making informed and confident decisions during these operations, the survey asks the participants about the feeling of being supported by the system. In addition, the participant's ability to confidently make decisions during the experiment is subjectively evaluated by the practitioner [45].

4.6.4 SUS - System Usability Scale

In addition to an evaluation looking at the different system components and usability goals, the SUS questionnaire is included in the survey. The 10 questions together with the 1-5 answering scale, presented in Fig. 2.12, is integrated into the survey. This part of the evaluation was included to highlight potential shortcomings in the system, to see how this AR system performs compared to other SUS tested systems, and to get a verbal score of the usability of the system. To read more about the questionnaire see section 2.3.3.

4.7 Outline of the experiment

Every participant takes part in the experiment one by one. As soon as they enter the door seen in Fig. 4.1, the experiment starts. It is conducted by going through the five steps listed below.

1. The participants should read the instructions in the "Participant information handout" found in appendix A.
2. If any questions occurred during the first step, the participants have now the opportunity to ask questions and test the experiment setup. The participants get 10 minutes to complete these two first stages.
3. Execute the three operations without AR.
4. Execute the same three operations with the AR system.
5. Finish the experiment by answering the experiment survey.

4.8 Discussion of the design, arrangement and conduction of the experiment

When deciding on how to set up the experiment to best test the desired capabilities of the AR operator interface, there are different decisions taken that together result in what's presented in the sections above. In this section, a discussion of the most defining decisions and situations is presented. It is discussed in this section when the experiment hopefully is fresh in mind and to reduce the length of potential back and forth jumping within the report.

To start off, it must be mentioned and kept in mind throughout the analysis chapter (next chapter) that every one of the participants is students at the same university and with a personal acquaintance to the practitioner. This might have influenced their responses to the survey, even though they all were told to be as honest as possible and that the responses would be anonymous. Because it was difficult to find anyone without any potential personal bias, there is no way to know for sure whether or not the responses were affected by this. The results presented in the next chapter will, therefore, be discussed based on how they appear without this in mind because, hopefully, any discoveries within these results will be independent of the choice of participants.

In which run of the operations should the AR system be used?
When the results are presented in the next chapter, it is important to remember that every participant went through the operations first without AR, then with. This could introduce a bias towards the AR system when looking at certain aspects of the experiment results, especially the quantitative. The participants could have become more prepared for what they were supposed to do in the second run because they have had one practical run through the operations with the non-AR system. But at the same time, in the second run, a whole new system and interface are introduced. This could kind of "overwhelm" the participants causing them to focus more on the added features and not so much on the fact that they actually have executed the same operations once before. It was decided to test the AR system in the second run much because of the thought of reducing the amount of information the participants had to process in the start. By first getting to know the operations through the non-AR test, and then add AR to the operations in the second run, participants could have more capacity to utilize the AR information. If the AR system was to be used in the first run, participants could potentially get a feeling for the depth of field in the scene. This could reduce the felt effect of the added depth information because after using the AR system, they could have become familiar with the length you have to push the tool to reach the objects in the scene when doing it the second time.
The discussed decision was clearly surrounded by trade-offs in both directions, however, based on the research questions purpose of testing decision support and depth perception capabilities of the AR system, the operations were executed first without AR than with. Another potential way of doing it would have been to alternate. Some participants start with AR and some without. Due to the size of the evaluation, the additional workload this

represents when generating the results, and the overall increased complexity it was chosen not to alternate. Therefore, in future work, it could be interesting to, maybe through a smaller test, alternate and try to measure the size of this bias and investigate if there even is any.

The process of deciding what object the participants were supposed to control in the experiment

As this experiment aims to simulate an operator operating the tool or right manipulator of a work class ROV during a subsea operation, a dummy ROV tool is required. To acquire this, the opportunity to execute some operations with a robotic arm was first investigated. However, it turned out the only option for a functioning robotic arm, possible to control with a joystick, would be available close to the deadline of this report. Following this outcome, other solutions were up for discussion. After previewing different approaches the one described in section 4.3.1 with the wooden beam had the greatest potential to be realized. With a manual tool like this, the maneuvering of the tool is solely up to the test participants them self, without the interference of any electrical systems. This means there will be no delay between a control input and the physical output, i.e., instant change in the position of the tool. On the other hand, the operator will be controlling the tool without the support and accuracy that an electrical system such as a robotic arm could offer. Due to this reasoning, the workload of the experiment should be more physically demanding than a system with an automated tool. However, it should be easier to learn and understand how to operate the tool because of its basic complexity and the fact that the user can get a physical feeling for the "controls" of the tool.

The decision to develop a tool from scratch rather than using something that already was developed is based on the freedom to design the operations and tools together, and thus create greater cohesion between the two. When deciding to go for a manual tool over an electric system, availability and complexity were the main drives. The sooner the tool was available, the sooner the work with developing operations and planning the experiment and evaluation could start. Performing an experiment with human test participants is a demanding task because everything has to be well planned and in place when the participant arrives. Thus the availability of a tool that could test depth perception and decision support was an important aspect of the decision. However, it is also more convenient to execute the experiment if the complexity is kept at an adequate level. Thus more time could be spent on developing the AR system, rather than deep dive into the nitty-grittys of controlling complex systems like a robotic arm or underwater drone.

The operations

When designing the operations executed in this experiment, the most important designing factor was to test the decision support and depth perception capabilities of the AR operator interface. This is why the first two operations rely on being at the right depth in the depth field before stabbing the tool and operation 3 is a more action based operation. Why the physical components in operation 1 and 2 are the way they are presented in this chapter, is solely down to decisions made in the designing phase when trying to create operation scenes which test the main designing factor and at the same time takes advantage of the tool design. There were plenty of different potential directions to go, but the end results provided the operation setup necessary to test the AR system following the purpose of this experiment. Moving the focus over to operation 3, where the concept in which this operation builds on is quite interesting. To have functionality in a system that supports the operators to check if the physical reading of the valve subsea is the same as the one listed in the digital system could be beneficial as the reliability of electrical and digital systems are not 100%. A valve with another position than actually intended could potentially have huge consequences for both the subsea production systems and the production plan. When the ROV is on its way to an operation or a mission, the ROV operator drives it often past several valves. By having a system working in the background (e.g., tracking system or computer vision system) checking these physical valve positions displayed on indicators on the subsea structures, the overall reliability percentage could be increased. That is because you suddenly could get confirmation if the reading in the digital production system is the same as the physical one at the seabed. If miss matches are detected, an AR information box could appear in the operator interface and alert about this situation. This is a bit out of the scope of this report, however, it is an intriguing idea that appeared when designing these operations and should be considered to be looked into in further work.

Chapter 5

Analysis

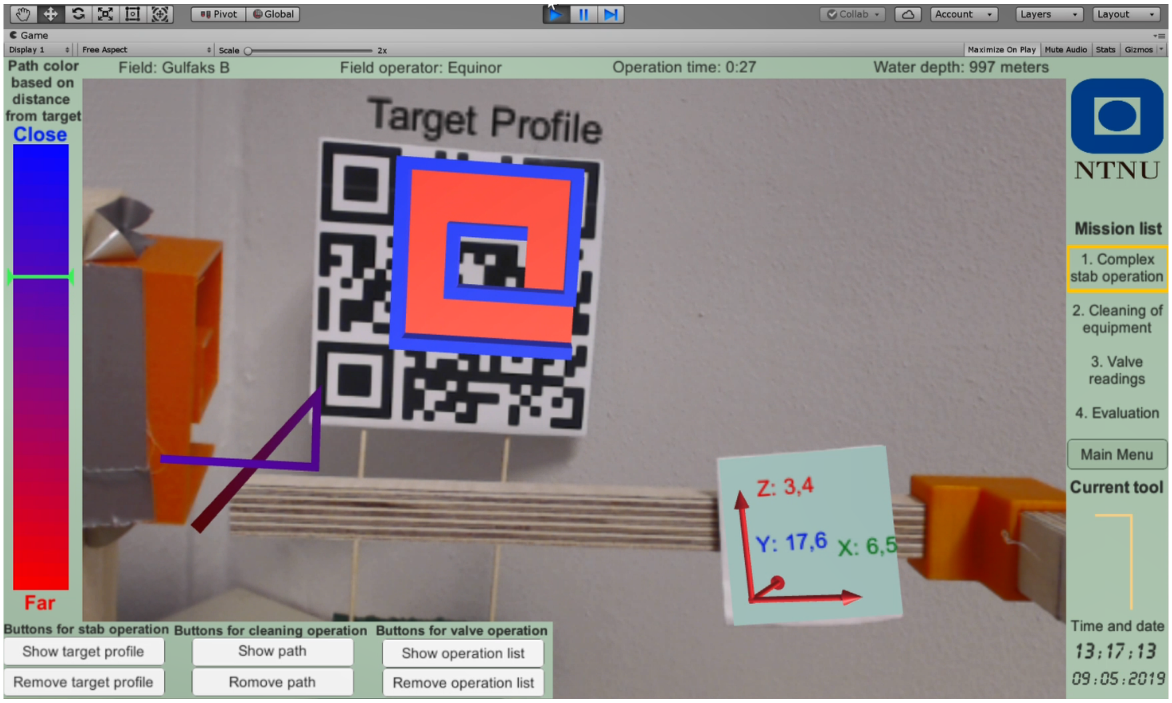
In this chapter the results will be presented in different stages, followed by a discussion of these results in every stage. The first results presented are the implementation results, which is the developed AR system. This system will also be discussed before moving on to the results of the conducted experiment. These results are divided into quantitative and qualitative results, presented in that order. The quantitative looks at what actually happened during the experiment (i.e., time used and amount of stabs in empty air), while the qualitative is based on the participant's subjective feelings after conducting the experiment. This evaluation is done in different stages, through an evaluation survey (section 4.6.1) generating general subjective evaluation, usability evaluation, and SUS results. The latter is also a form of usability evaluation, however, it is presented separately to keep the focus towards the defined usability goals in the main usability evaluation section. An important remark for this chapter is that the user of the AR system will in the first sections when looking at the implementation, be referred to as "the operator". In this case, it referees to subsea ROV operators, while in the sections based on the experiment the user will be mostly mentioned as "the participant". If participant is used, it is to describe something taken from the execution of the experiment.

5.1 Presentation and discussion of the developed AR system

In this section, the results from the implementations in section 3.3 will be presented and discussed. This is the developed AR system, which is the AR operator interface that visually consists of the different depth tools and operation-specific AR information. The depth tools are developed in an attempt to give the operator (participants of the experiment in this case) depth perception while looking at a 2D screen. The operation specific components, on the other hand, is there with the main goal of supporting the operators through decision-making processes during the different operations.

5.1.1 The AR operator interface

From the implementation described in section 3.3, an AR system was developed resulting in the interface shown in Fig. 5.1. As shown, the interface has a layout with green borders around the edges. These borders present different operational information to the operator and are called the general UI. The right border includes an informative mission list which lists all of the tasks within the current mission, time and date, and the specific tool equipped on the operated vehicle. On the top, information about the current field is presented, together with the time spent on the current operation task and the water depth of the vehicle. A reason for including this border information is to make sure the AR system is developed within the requirements of the NORSOK U-102 standard [106]. Lastly, on the bottom, there are buttons which let the operator turn on and off certain AR components in the scene. This lets the operator remove components if they should be hindering important line of sight or if bugs appear. In addition to these borders, there are three different components giving the operators depth of field information, i.e., depth perception. These are the depth bar, which is at the left side of the general UI, the path between the tooltip, and the target and a 3D coordinate system which offers 3D information through dimensional data presented in centimeters. In addition to this, the system also consists of the different AR components presented to the operator during each of the three operations in the experiment. These are the components introduced in section 4.4.



Field: Gulfaks B

Field operator: Equinor

Operation time: 0:27

Water depth: 997 meters

Time and date

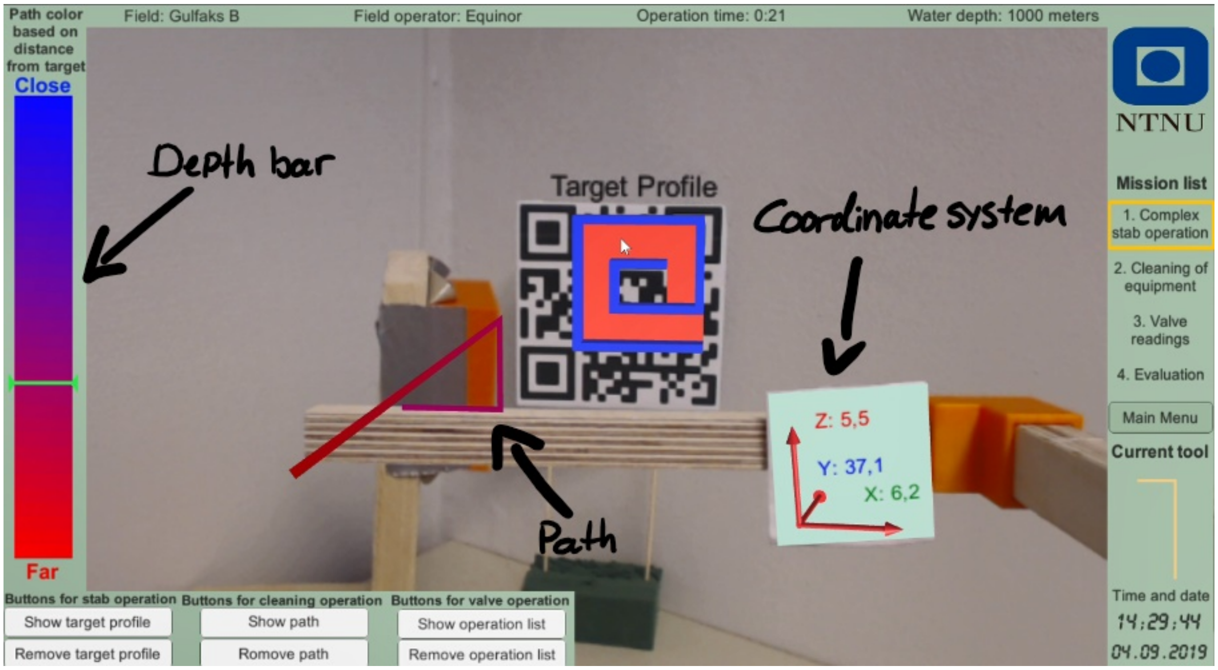
13:17:13

09:05:2019

Figure 5.1: The developed operator interface during operation 1 of the experiment.

5.1.1.1 AR depth tools

With the research question of this report in mind, the AR depth tools are the main spine of this developed system. They are here in an attempt to solve a huge task, which is to give the operators 3D directional information while looking at a 2D object (e.g., a monitor or a tablet screen). By utilizing the capabilities introduced with Unity and Vuforia both the visual frontend and the backend works together to produce the resulting depth components presented in this subsection.



Field: Gulfaks B

Field operator: Equinor

Operation time: 0:21

Water depth: 1000 meters

Z: 5,5

Y: 37,1

X: 6,2

Time and date

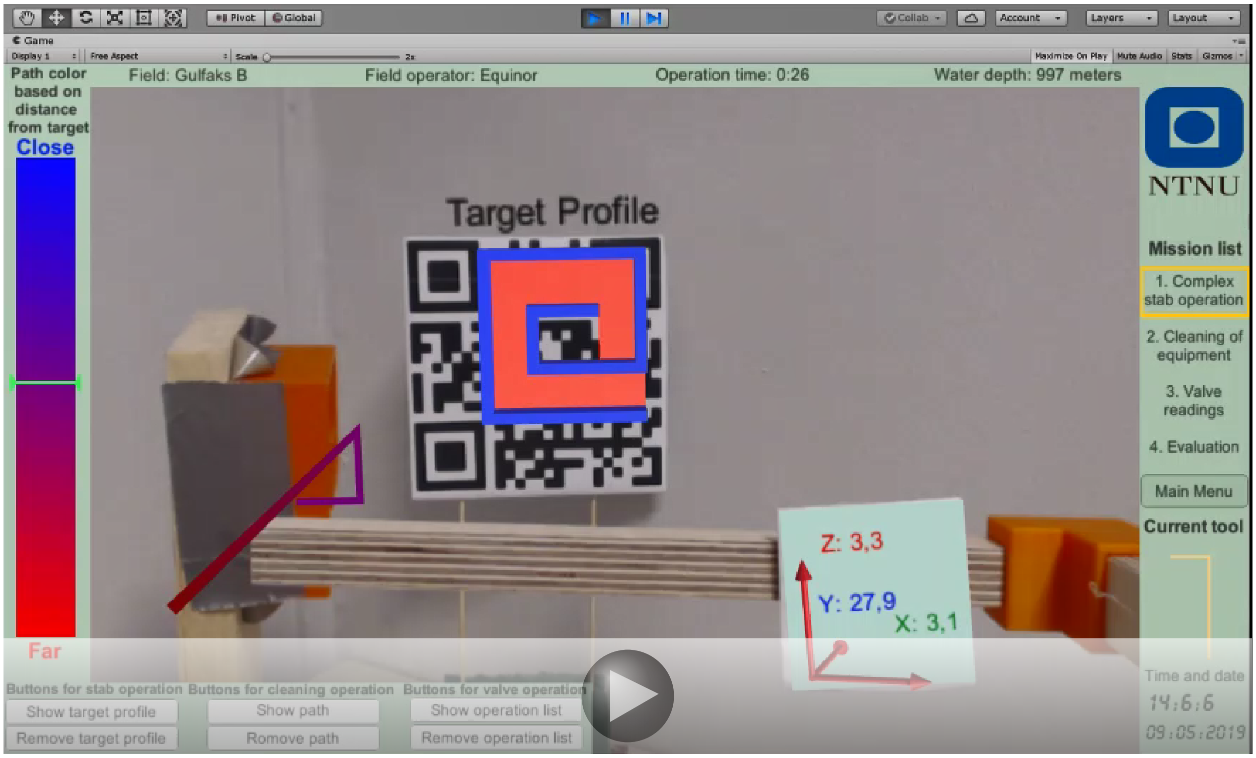
14:29:44

04.09.2019

Figure 5.2: Illustration of the depth tools in the AR operator interface.

Path

As mentioned above, the path is one of the components presenting the operator with depth information. The path is indicated in Fig. 5.2 and in Fig. 5.3 it is possible to watch how the path indicates a potential route towards the target in three dimensions. It starts at the tooltip and ends at the target. To reduce the impact of noise in pose estimates there is an averaging filter taking the average of the last four estimates before that averaged estimate is assigned as the path value. The path also changes colors when the tool tip's position is changed. This color change follows the color scheme of the depth bar, i.e., going from red when far from the target to blue when close to the target. This way, the depth information is given to the operator in two visual ways, through changed line lengths and changed colors.



Field: Gulfaks B

Field operator: Equinor

Operation time: 0:26

Water depth: 997 meters

Z: 3,3

Y: 27,9

X: 3,1

Time and date

14:6:6

09:05:2019

Figure 5.3: If clicked plays a video showing the different depth tools in action during one of the participants run through operation 1 with AR. It should be noted that the date and month got switched up during the two days of the experiment. It should be 05:09:2019, not 09:05:2019.

Coordinate system

The developed AR Coordinate system gives the operator in real time updated pose estimates of the camera relative to the tracking marker in the scene. It is placed on top of the marker on the tool which is "hidden" behind the green background, as shown in Fig. 5.2. With the help from real-world dimensions of the tracking marker in the scene, the system manages to present the x, y, and z-distances in centimeters. Notice at the coordinate system in Fig. 5.3 how the values are changing in real time and approaches zero as the operator maneuvers the tool towards the target.

Depth bar

Another of the depth components is the depth bar. With the development and implementation described in section 3.3.2.2 the resulting depth bar placed to the left in the operator interface, as seen in Fig. 5.2. The bar is also shown in live action in Fig. 5.3. In this video, one can see how the slider moves towards the blue as the operator maneuvers the tool closer to the target. At the same time, if the operator moves the tool too far so that it ends up behind the target, the slider will indicate that the operator is as close as possible to the target, and not behind it.

5.1.2 Operation specific AR information

In addition to the depth tools, there are several other AR objects in the system. Some of these are specific for executing each operation, while others (the general UI) is standardized for use during every operation.

Operation 1
In Fig. 5.4 and video in Fig. 5.3 the resulting AR system during operation 1 is presented. It shows how the AR maze-profile (i.e., target profile) is integrated into the scene. It is placed conveniently to help the participants maneuver the tool through the 3D printed maze. In the general UI, the right border shows the L-profile tool as "current tool" and the corresponding operation is highlighted in the mission list.

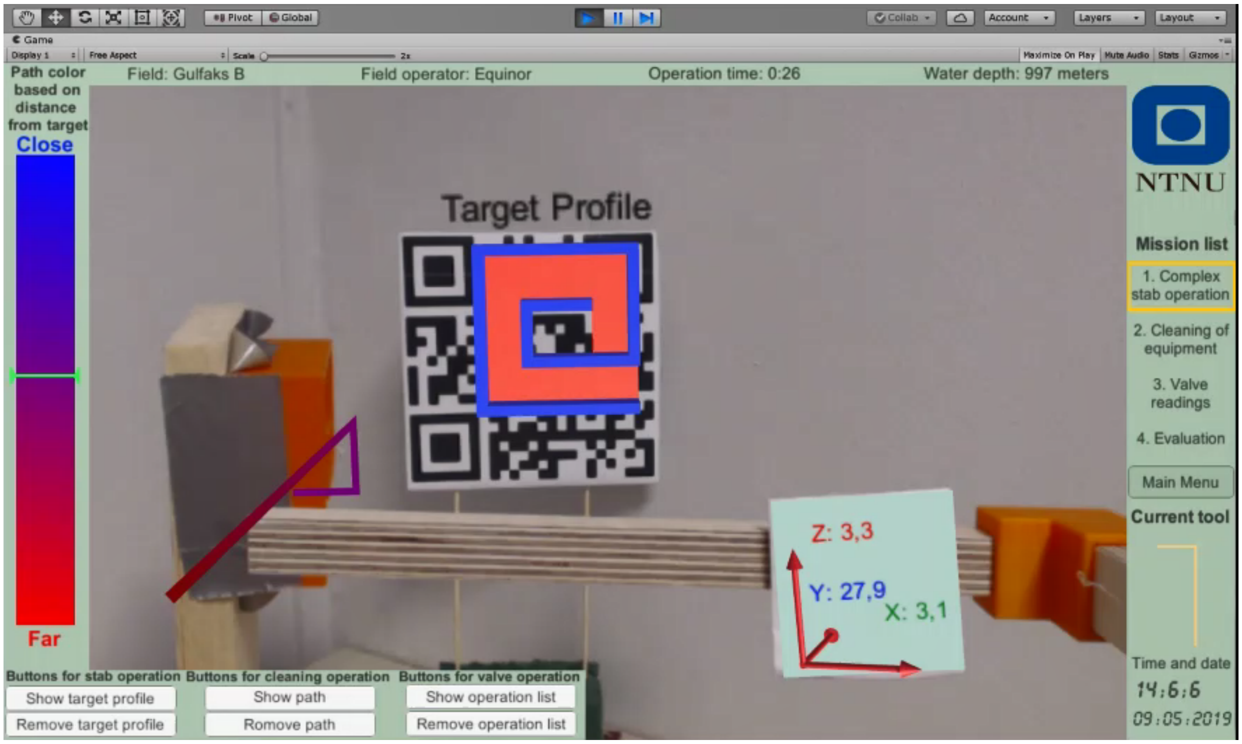


Figure 5.4: Illustration of the AR system during operation 1.

Operation 2
Moving on to operation 2, the AR specific component is the indicator shown in Fig. 5.5, if clicked shows a video of the operation. This component is created to highlight the placement of the quadratic hook and help the participants remember to pull the tool down to remove the dirt, this means it is a down-pointing indicator serving two purposes. The indicator is also shifted slightly to the right in order to indicate what side of the quadratic hook the tool is supposed to enter from. Not visible in the figure is the 3D text which appears above the cleaned marker when this is fully cleaned, the component is, however, illustrated in Unity

in Fig. 4.16. Between operation 1 and 2 there are no significant changes in the general UI other than which operation is highlighted in the mission list.

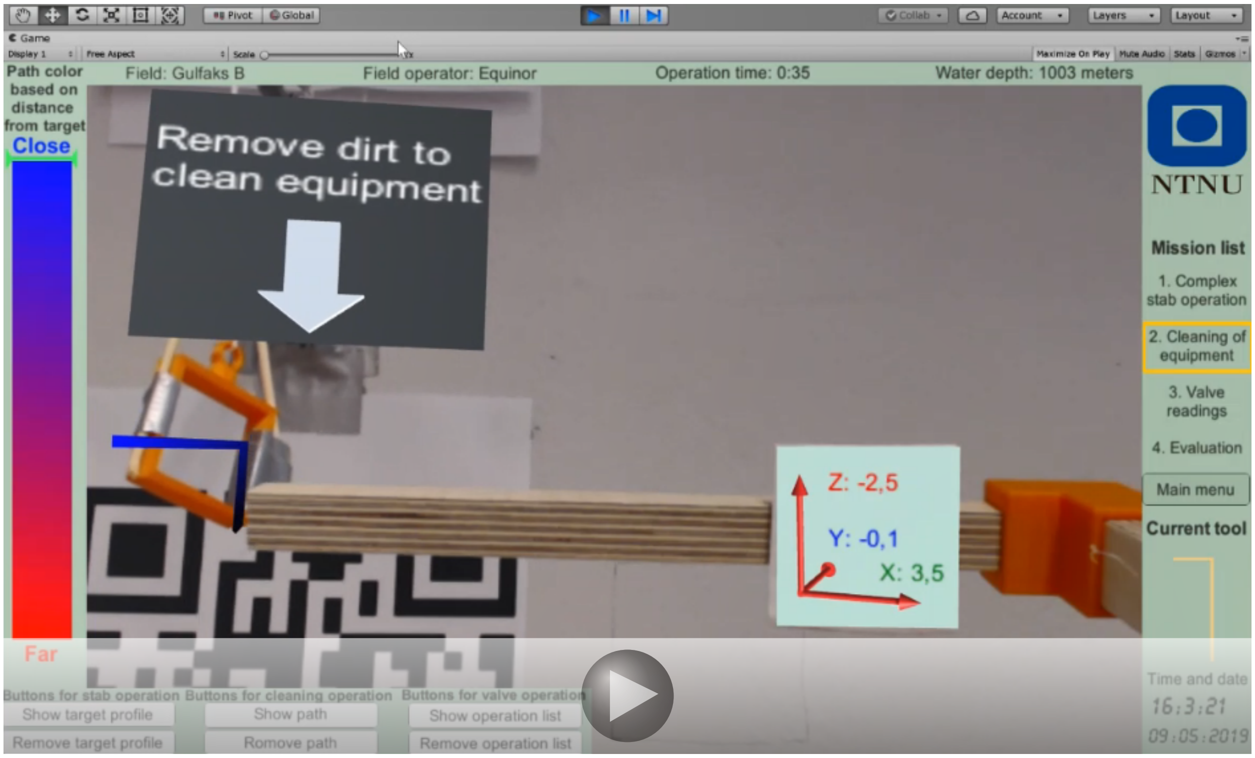


Figure 5.5: If clicked plays a video showing the execution of operation 2 with the AR system.

Operation 3

Going to operation 3, the tool has now changed. The horizontal bit is detached, and at the same time is the tool shown under "current tool" in the general UI also changed. This is illustrated in Fig. 5.6, which also shows the operation specific AR components for operation 3. Here are the valves presented as AR information with the intention to be more visual for the participants compared to the physical dummy valves at the wall. In addition, the printed handouts provided to the participants during the run with the non-AR system (see Fig. 4.20 and appendix A), is now integrated into the AR system. This results in a live-updating operation list, who changes the displayed information as the time changes so that it always presents the relevant valve positions. An updated operation list for both valve 1 and 2 are thereby presented right above the AR valves. The remaining information from the handouts is presented and utilized when going through the button sequence on the left side in the operator interface. Through this interactive sequence and the displayed information, the participants are supposed to have everything needed to make educated decisions regarding the valve position to complete the operation. To see the button sequence in action, click Fig. 5.6 to start a video showing one of the participants completing operation 3.

Field: Gulfaks B

Field operator: Equinor

Operation time: 0:4

Water depth:

Do you want to change the position of any of the valves?

Operation List - Valve 1
09.05.2019
- 15:00, chocke: 50%
- 16:00, chocke: 60%
- 17:00, chocke: 60%

Operation List - Valve 2
09.05.2019
- 15:00, chocke: 60%
- 16:00, chocke: 50%
- 17:00, chocke: 40%

16:13:10
09:05:2019

Figure 5.6: If clicked plays a video showing the execution of operation 3, by a participant, with AR.

The general UI

Some of the dynamic elements of the general UI have been mentioned above under each of the operations. It has, however, some additional features. It features small information areas both in the top border and the right to make the AR system better cope with specific ROV operator interface requirements. There is also an area containing off and on toggles of certain AR objects, these areas are all indicated in Fig. 5.7. In addition, it has a main menu button to help the participants change the operation mode of the AR system during the experiment. Another purpose of the general UI is to make a similar layout during every operation to create a familiar setting for the operators and give the system a distinct and recognizable character.

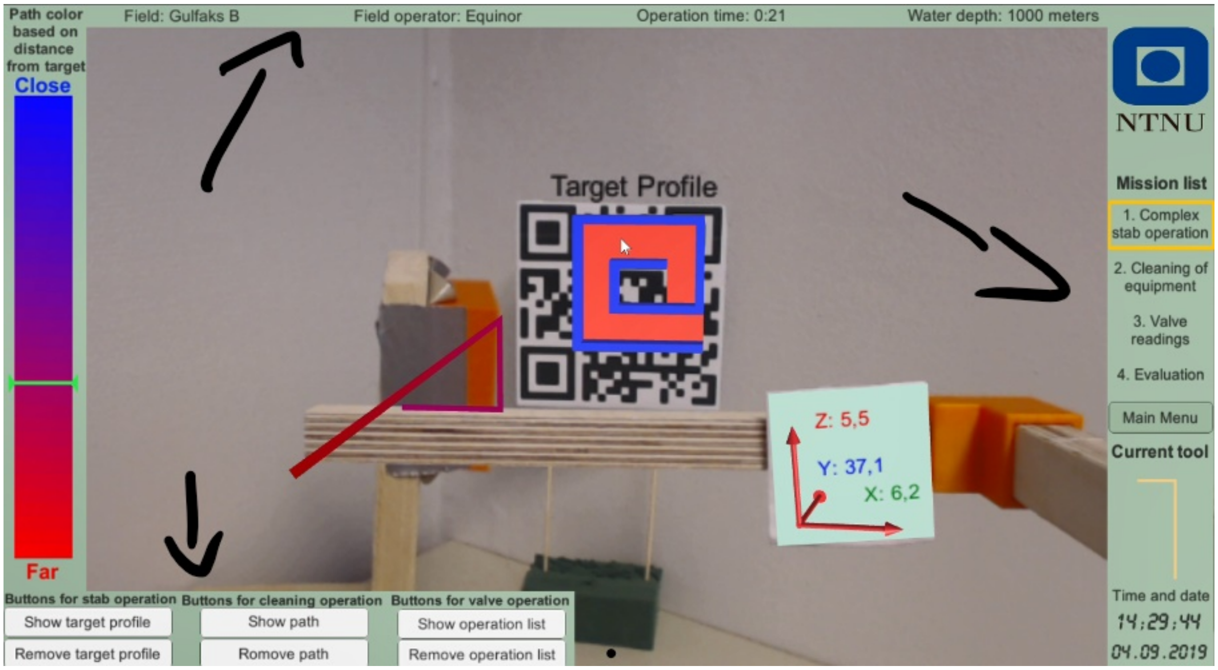


Figure 5.7: Illustration of the different borders in general UI.

5.1.3 Discussion of AR system

By looking at the implemented system, there is no doubt that it is a system consisting of AR objects. How the system is performing and providing depth perception and decision support capabilities is to be discussed in this section. This is indirectly done in this section by looking into the robustness and reliability of the integration of the AR information.

Robustness and reliability of tracking

When deciding and designing the AR objects which are supposed to give the operator's depth perception and decision support, there are a lot of potential directions to go. Each of the three AR depth tools (section 3.3.2.2) presents the operator with depth information in different ways, these are differences in components, design, color, and texture. All of these factors have a saying in how the user perceives the AR information[107]. In addition to the design and form factor, the placement within the operator interface is also of importance. This is at the same time one of the big challenges when developing an AR interface. Because placing virtual objects so that they align with objects in the real world is not a straight forward task [108]. This is also vital when trying to make the virtual AR look as realistic as possible[109]. These challenges were introduced in section 2.1 and will play a central role in the discussion in the following paragraphs.

By looking at the general robustness of this marker-based AR system, the Vuforia engine seems to do a solid job of keeping track of the markers and overlay them with the intended information. It actually performs rather good even during rapid turns and changes in position. This tracking robustness is probably the main advantage of utilizing the powerful

Vuforia engine over some self-developed tracking system as suggested in section 3.2. However, it does not come without disadvantages and limitations. First of all, the markers had to be rather large for the system to detect them, as described in section 3.1. They were clearly visible in the camera, but the tracking would simply not start before moving closer to the markers. It might, therefore, be a built-in tracking performance criterion needed to be surpassed before the engine presents the tracking to the developer, in order to ensure robust tracking. This information is not open to the free world, as the code is not open source, and is solely based on experience when developing the system. The only provided information from Vuforia is the rule of thumb that the marker should be ten times smaller than the distance between camera and marker [68]. This made, however, the markers a substantial part of the visible scene and not some discreet objects in the background, which it first was intended to be. Secondly, the engine had troubles with removing AR components from the scene when the marker was no longer visible. Leading to AR objects drifting about in the scene. This was luckily fixed with some modifications in Unity, but it does, however, limit the overall reliability of the system, forcing the implementation of on and off buttons in the general UI. These were also needed as occasionally bugs in marker tracking occurred and thus messed up the placement of AR objects. A restart (off and on toggle) was often enough to get them functioning properly again. A video showing this is found in the enclosure with the file name "ARfloat example". These reliability issues also forced us to segment the AR system into three different Unity scenes, one for each operation, as AR objects from the prior scene sometimes drifted about in the scene during the next operation. It was first intended for the AR system to seamlessly work from operation 1 to 3 in one go. But with this split, a main menu was developed, seen in Fig. 5.8, where the participants can choose between the different operations and the non-AR mode. The participants, therefore, had to go back to the main menu after every operation was complete to select the next one. Why these issues occurred is still not discovered. Perhaps the markers from the different operations were too close? Even though markers from other operations were not visible in the video feed during a specific operation. Or the integration of Vuforia in Unity might not be perfect yet and this is one of the implications? They have, however, been official partners for about 2 years now, and the Vuforia engine has been integrated as a part of Unity since October 2017, before this you had to download it as a separate developer kit [110], so one should think that potential integration problems should have been resolved by now, if there even were any. The problems were, however, worked about with the split, but it has a negative impact on the robustness and the experienced operation flow of the system.

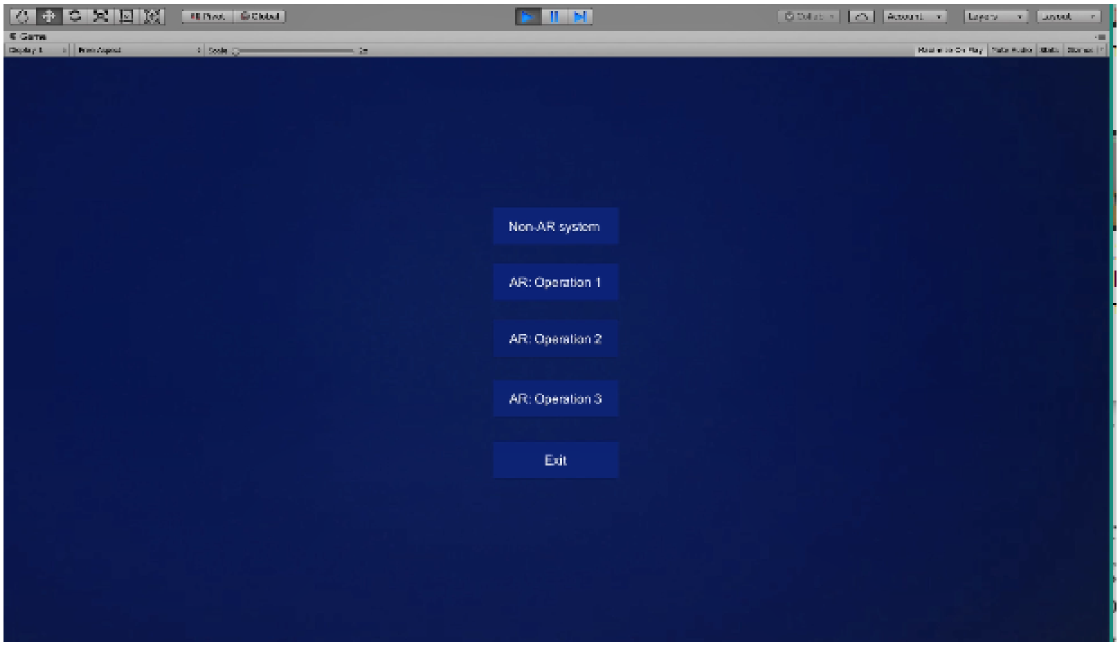


Figure 5.8: Illustration of the main menu of the developed system. Its C# script can be found in appendix C.

As described in section 2.1, one of the main challenges of implementing AR, is to make sure that the process of integrating virtual objects in the real physical world happens as seamlessly as possible. This was probably the most time-consuming part, besides writing the code in the C# scripts, when developing this system. When using marker-based AR, the system doesn't fully understand the world captured by the camera. It only understands parts of it, which is the markers placed around in the scene. The process of adding AR components directly on top of these gets rather trivial once you get into the flow of implementing AR. Another thing is to align components with non-marked objects in the scene. An example of this is the path. It originates at the tip of the tool and ends at the target within the scene. This component relies on pre-measured distances between markers and the desired origin and end-point. The scene was only calibrated once, thus not before each experiment run, and the alignment of, especially the path, is very dependent on the camera being placed exactly the same way in the docking as during calibration. Because the camera has its own mount with an adjustable joint that tilts the camera up and down, there is a potential that the camera angle has been changed during the course of the experiment. This is most probably the reason why the origin of the path and thus also the endpoint is not perfectly aligned, something that is illustrated by the different figures under operation 1 and 2 above. This might limit the user experience as well as the AR components versatility because it can only be used in that specific setup unless new calibrations are made. However, besides the camera angle uncertainty, it is sufficient to use in this case in order to investigate the research question (presented in section 1.2).

For this to work in a non-constructed setting, the AR system could benefit from a markerless setup. With this, the system could potentially be able to identify the origin (tooltip) and the desired target by itself, thus managing to calculate the distances and align the path in-between these two points on its own. This is though, time-consuming as it requires a good performing computer vision system, which understands every bit of information about the environment captured by the camera, as mentioned in section 2.1. Such a system would highly increase the robustness and versatility of the system as it could have a much greater chance of succeeding in a variety of scenes, not only limited to the one calibrated for. This is, therefore, probably the biggest limitation of this system, as some of its components only perform as intended in the strategically calibrated experiment scenes. An interesting method to look into with the marker-based system is, however, the introduction of sensor fusion. If the system was given additional information to help estimate the position of the camera and markers from the introduction of different sensors. Then with the help of sensor fusion, it is possible that the system could manage to improve the alignment of AR information with the real world [111, 112]. Especially during fast movements and fast changes in the scene. When relating this to the work class ROV case, different types of position sensors including IMU are already in place on the vehicles, which essentially means this method probably won't require additional equipment or sensors.

Even though the system is limited to function in pre-calculated scenes, a way of increasing the robustness could be to place more markers in the scene and place them directly at the targets. If this was the case the depth perception capabilities could be used in an arbitrary scene, as long as the distance from the target to the tooltip is known through markers. This could, therefore, be used for different purposes during subsea IMR operations. With markers placed around on subsea structure that the ROV interacts with frequently during IMR operations, this system could potentially give information about the distance between the ROV and the subsea structure to the operator. Only this simple depth information could come a long way in giving the operators some sense of depth perception from the 2D monitor. This distance information is of importance to the operator as the left ROV-manipulator is designed to be used to grip on to the handlebars on the subsea structures, as described in section 2.2.1. A task that could be executed very comfortably with depth information, compared to without.

There are of course many aspects to think of before pushing this system subsea. Maybe the biggest task is to make sure that the marker is clean and visible at all time. Other things to take into account is the turbidity of seawater, disturbances from life underwater (e.g., fishes), loss of color, etc. These factors were discussed in our earlier report [12]. It is also worth mentioning that interesting work is done in the development of systems tracking markers under water [113]. Such a system could potentially replace the Vuforia engine, as this is not an engine designed to perform in harsh underwater conditions.

From this discussion of the robustness and reliability of the developed AR system, it should be said that despite the limitations, when the system actually runs and works within its designed testing area, it does its job of providing depth information to the operator. That does not mean the depth information provided is accurate down to the finest of millimeters and that the AR components are perfectly integrated. It only states that the system manages to inform the operator of the desired information this report sat out to provide, at least one part of it. How the other aspect of decision support is handled and more details about the depth information are to be discussed throughout the remaining of this chapter.

Depth tools

Moving the focus over to the three depth tools. First of all, it should not be necessary with three different tools. They all represent the same information, the difference is though in the way they present that information. Without any tests and evaluation feedback to rely on in the designing phase, it was difficult to decide which to go for, therefore three different depth tools are included in the system. Because a part of the goal with this report is to look at how depth perception can be implemented, it will be interesting to see how these three components are evaluated. However, the results from this evaluation of these tools and a discussion do not come before the next couple of sections. Here, we are going to discuss how they were implemented and how they appear in the AR interface.

As mentioned earlier, the path is a difficult part to get working. Different from the two other depth tools, it is not anchored to a marker or the general UI. It is, therefore, more difficult to align the AR effects with the real world. This was a bit problematic when trying to align the start of the path with the tooltip. As seen in Fig. 5.1, where the origin of the path is not perfectly aligned. At first, it was tried to align this AR effect by only using the main marker placed in the background of the operation scenes. It turned out to be difficult to execute in practice as the depth alignments got screwed up. A solution to this became to add another marker, this time directly on the tool. This way, the origin of the path could, depth wise, be at the same depth as the tool marker only shifted towards the left to meet the tip of the tool. Ideally, to place the path even easier onto the tooltip, a marker could be fitted at the tip of the tool. This could have been possible if a marker as small as the wooden beam itself was sufficiently large.

But the smallest sized marker possible to use, based on the equation in section 3.1 and the camera placement, is about 3-4 times wider than the tool beam. It would thereby be very inconvenient during the actual stabbing of the operations to have a marker of that size at the tip. Important line of vision would also most definitely be blocked by it. For this to work with a beam sized marker, the camera would have to be way closer to the horizontal beam, which would narrow the captured field of view. On a positive note, the addition of a marker at the tool made way for convenient placement of the coordinate system as it follows the tool around in the scene without blocking more vision than the empty marker behind would have blocked.

Another of the depth tools is the depth bar. First of all, it is worth mentioning how the slider range, i.e., the distance from the reddest area at the bottom to the blue at the top, is pre-defined. It is set to range from 60 to zero, with zero being when the tool is at the target and 60 at the reddest area. This works fine in this experiment as the tooltip cannot be moved more 60 centimeters away from the target, due to constraints from the wall behind the operator and the length of the beam. But in another setup or another use case, the depth bar would rely on some distance references in order to achieve the same functionality. Say, if this were to be used during subsea IMR operations, it would probably be cleaver to include measuring references as you find on a ruler. Only with meters rather than centimeters.

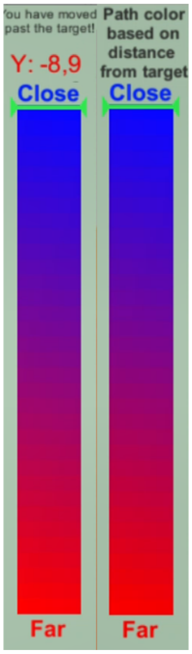


Figure 5.9: Improved depth bar.

Another, potentially bigger limitation, is the fact that if the tooltip is maneuvered behind the target, the slider has no way of informing about it. The bar simply displays that the tip is as close as can be to the target, even though it actually has passed it and is behind the target. This could potentially limit the decision support aspect of this depth tool as it cannot help you decide whether or not you are at the target or behind it. A way to improve this depth bar could potentially be to add the Y-direction information from the coordinate system to the depth bar when the tool has passed the target. A suggested improvement is shown in Fig. 5.9, where the new depth information is added to the bar on the left side when the Y-value turns negative.

Field of view, size, and location

One of the important things to keep in mind when adding objects to an already limited field of view, due to camera's more narrow field of view compared to humans, is the placement of each component. This is a difficult part when developing an operator interface and it could potentially make or break the experience for the user. At the same time can strategical placement of objects help draw attention to the most important pieces of information [92]. With this in mind, the overall design of the system ended up like what's illustrated in Fig. 5.1. It is a result of the trade-off, of trying to provide as much information as possible to the operator while trying to not overcrowd the operation interface.

The visibility and space are reduced because there are three components providing depth information, rather than more ideally just one. A focus area during the design process was also to create a general purpose layout which is supposed to have much the same appearance during every operation, but with some changes in the information it displays. This is in order to make an interface that operators can familiarize them self's with, and thus hopefully easily understand where to expect to find relevant information. This is an important aspect of user interface design, as a standardized appearance of the system during different operations could enable the operators to almost automatically respond to different situations. That is possible because they after time have gotten used to where to find the information or functionality needed [114, 115, 116]. The resulting general UI (the green information bars) takes a substantial amount of space in the already limited view, but it is placed out along the edges of the screen not to hinder important line of sight during operation. It does, however, limit the general view of the surroundings.

When looking more into detail on the depth tools placement in the operator interface, the placement of the path is right at the center of the operator's visual attention. It is right where the action is expected to happen, as it finds itself right between the two objects which are supposed to meet. This is one of the more important aspects when developing user interfaces, to place frequently used information conveniently and close to or within the user's main focus area on the screen [117]. In a general setting, when having the path right at the center of this focus area, this should limit the number of times the operator has to shift the focus elsewhere on the interface when navigating to the desired destination. Because the horizontal beam of the tool is used in two of the operations in this report, the center of focus is should be placed further left than the center of the image. The participant's visual focus when conducting the experiments should then naturally shift to the left, towards the tooltip. Something which could benefit the depth bar, as its placement at the left side in the UI appears to be close to the participants focus area. This is great if using the depth bar since the operators won't lose as much awareness of what happens in the area around the tooltip. The losing part is probably the coordinate system, as this is placed more to the right and thus on the opposite side of the main focus area. But the fact that the coordinate system is placed on top of a marker, introduces information in an area that else would have just been a randomly placed QR-code. This helps positively on its placement in the operator interface, seen from a designers perspective. It does, however, also benefit the operator, as the coordinate system follows the movements of the tool, ensuring that it is always somewhat in a close approximation of the main action and focus area.

As for final remarks, the AR operator interface has been implemented with both tools and components which on paper is supposed to be capable of supporting operators both with decisions and depth perception. It has been discussed how some roadblocks have been worked around in order to present the results discussed in this section, and the motivation behind the placement of certain elements in the interface. It also appears that the process of integrating the AR layer onto the video feed from the camera has been one of the more challenging processes of this implementation. The limitation of how the AR system is calibrated to work in the experiment setup used in this report is also mentioned, however, the system does only need a re-calibration to function like intended in a different camera setup in another operation scene, as long as the scene has markers in it.

5.2 Quantitative experiment results and discussion

During the experiment, some quantitative results were collected in addition to the qualitative from the survey and evaluation. Generally, the quantitative results say something about what actually happened during the experiment. In this case, the time used to execute the operations and the number of stabs in the air (see section 4.6.1) before finding the correct depth to execute the operations are the quantitative results. These results were timed and counted by the practitioner and collected through the survey described in section 4.6.1. The data presented here is converted from Google Forms [104] to Microsoft Excel [118], where they are processed to get the appearance presented and discussed in this section.

5.2.1 Time spent on each operation by the participants

As mentioned above, the participants were timed when executing the different operations in the experiment. They were, however, told not to stress or worry about the timer, because the purpose of the experiment is not to execute it as fast as possible. The actual purpose is described in section 4.1.

Operation 1
The graph in Fig. 5.10 displays the time used by every participant to complete operation

1 both with and without the AR system. Immediately, one can see that every participant, but one, complete the operation faster with AR compared to without. The one participant standing out had some technical problems at the same time as the participant did not fully understand how to execute the operation. In the run with AR, the person was inspired to simply stab the tool through the hole at the center without going through the maze. The total time accumulated when this person had to move the tool back and start over. Other than that, most participants improve their times significantly with the AR system. The greatest improvement is participant 1, who improves with a margin of over 50%. Looking at these results gives an indication that the participants had great help from the depth information and the AR system when executing this operation.

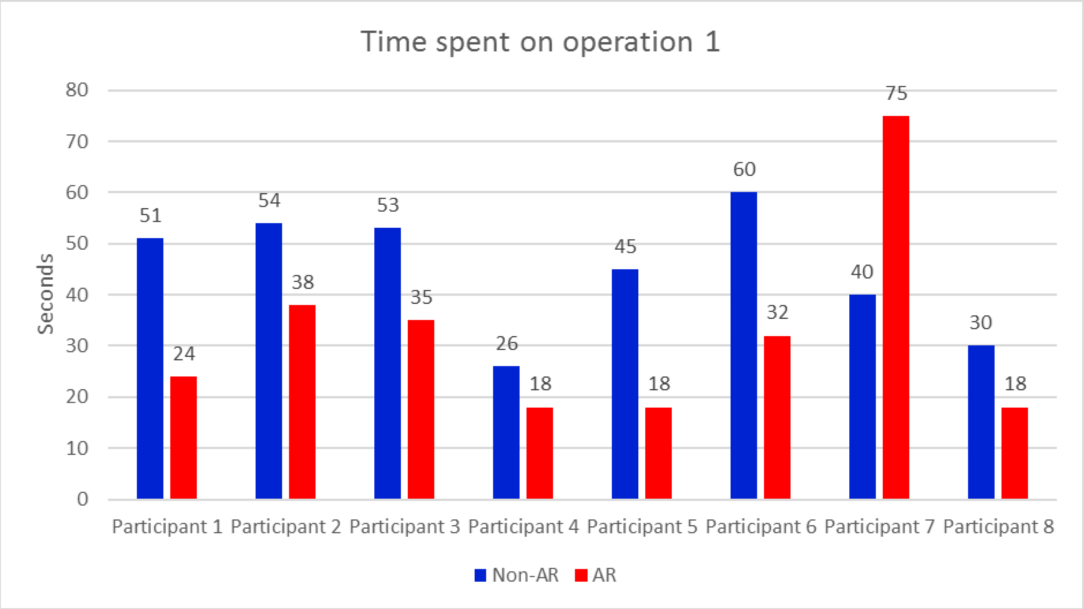


Figure 5.10: Time spent on executing experiment operation 1 both with and without the AR system.

Operation 2

Moving on to operation 2. In Fig. 5.11 the timed results are presented in a similar graph compared to the one above. In this situation, every participant completes the operation faster with the AR system. The participants manage this with a decent margin, ranging from 17% as the smallest to 50% as the greatest (achieved by participant 4). It is also interesting to note how three of the eight manages to execute the operation with the fastest time of 15 seconds with the AR system. Similarly, as for operation 1, it appears that the AR system aided the participants to cut operation time also in operation 2.

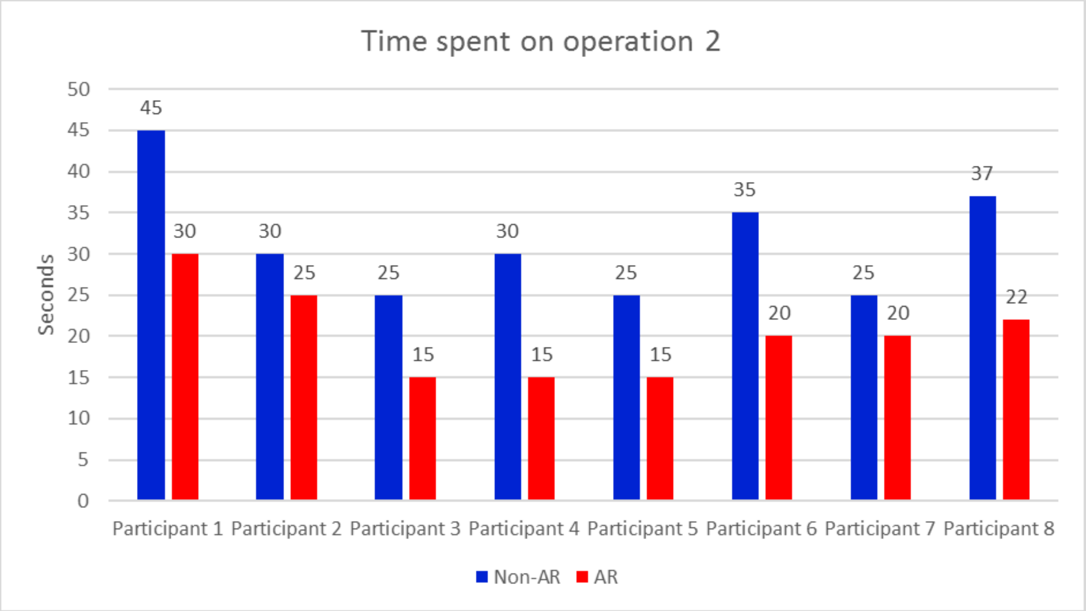


Figure 5.11: Time spent on executing experiment operation 2 both with and without the AR system.

Operation 3

For operation 3 we see a huge decrease in operation time when using the AR system in Fig. 5.12. The most significant reduction in time is participant 3 who reduced the time used by 87%. In addition, most of the participants reduced their time with more than 50%. Based on these differences it seems like the participants were greatly supported by the AR system while going through this operation.

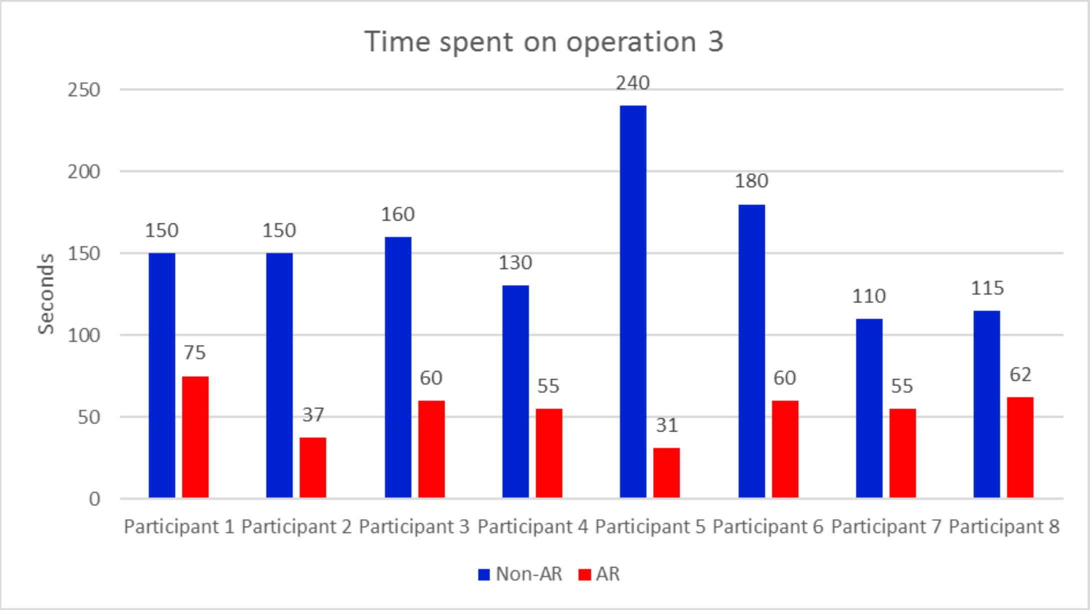


Figure 5.12: Time spent on executing experiment operation 3 both with and without the AR system.

5.2.2 Amount of horizontal stabs in empty air during operation 1 and 2

For the two operations where precision were required to stab the the horizontal beam of the tool inside a target, i.e., in operation 1 and 2, the amount of stabs in empty air was counted. In short, this kind of stab could indicate that the participants believe they are at the correct depth to execute the actual task within the operation, but because of the emptiness really isn't. For further explanation and demonstration of the stab, go to section 4.6.1. It is worth mentioning that the participants was never aware that the practitioner counted the number of stabs used to get to the correct depth to carry out the task within the operations.

Operation 1

In operation 1 one can see from Fig. 5.13 the spread in amount of stabs without the AR system. On average every participant does 5.6 stabs before completing the operation without AR. Following this average is the standard deviation amongst them of 2.12 and variance of almost 4.5. This puts a number to the varieties around the average which, varies around to the amount of stabs of the three equally stabbing participants with 5 stabs each. These values are calculated in MS excel [118] and listed in Table 5.1. With the AR components active, the numbers is not so much the same. The average has now decreased to 0.4 with also significant decreases in SD and variance. In other words, most participants do not stab in empty air in order to find the maze entrance while using the AR system. This is also illustrated in the figure by the lack of red bars.

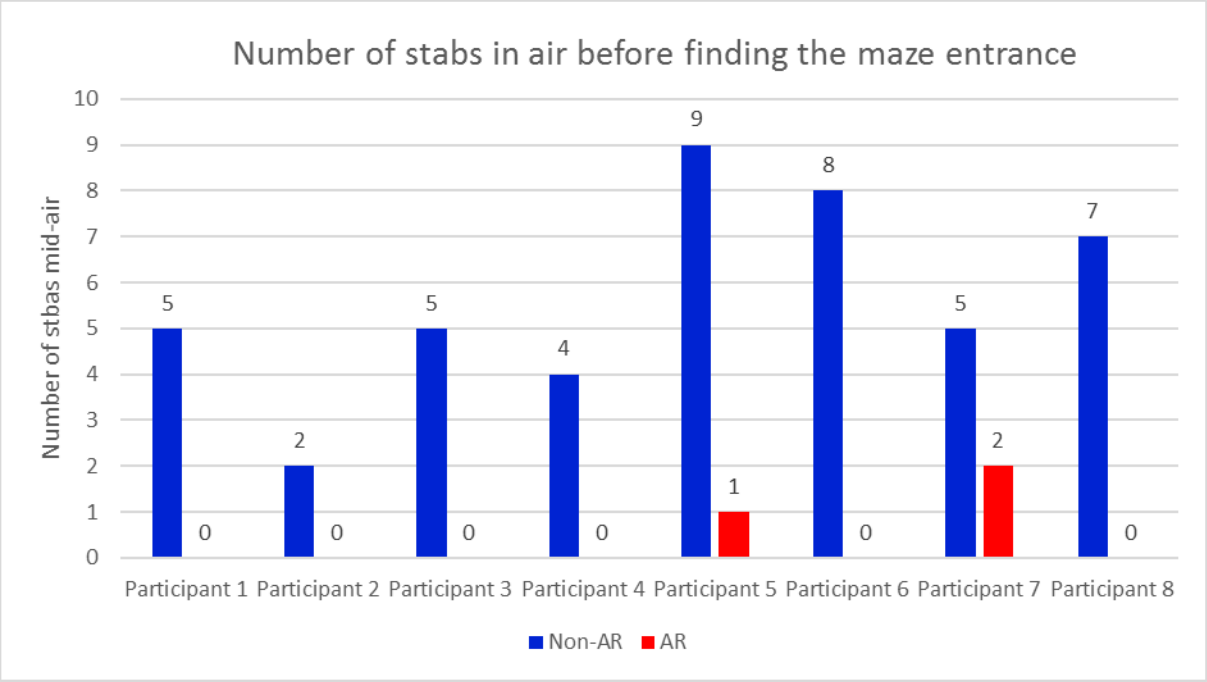


Figure 5.13: The amount of horizontal stabs in empty air with the tool for each participant both with and without the AR system in operation 1.

Operation 2
Moving over to operation 2 and Fig. 5.14, the same trends as for operation 1 is visible, however, a bit toned-down. On average participants does fever stabs in empty air without AR, 3.8 in fact. The deviations and variance is also a bit lower when looking in Table 5.1 and compare with operation 1. This trend is not the same when looking at the execution with AR. In this operation, half of the participants stabbed once into emptiness with the AR system active. Still, every participant did fewer stabs while using AR compared to without.

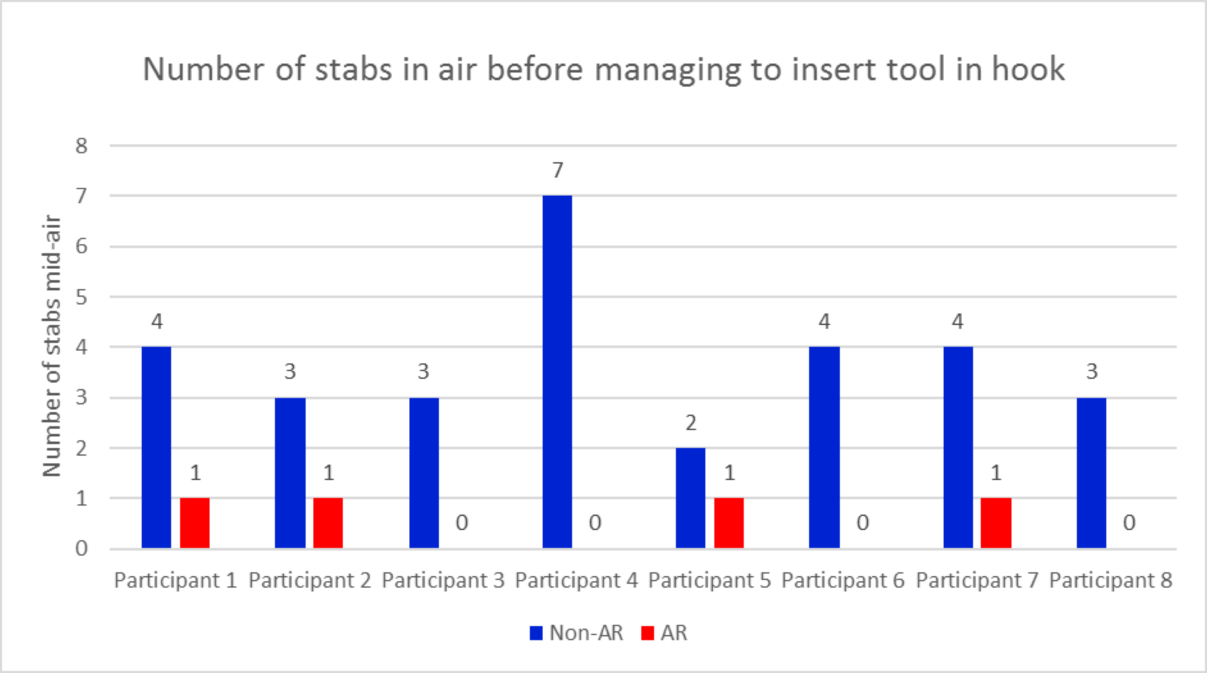


Figure 5.14: The amount of horizontal stabs in empty air with the tool for each participant both with and without the AR system in operation 2.

Operation	System	Avg	Var	SD
1	non-AR	5.6	4.48	2.12
1	AR	0.4	0.48	0.70
2	non-AR	3.8	1.94	1.39
2	AR	0.5	0.25	0.50

Table 5.1: The average (avg), variance (var) and standard deviation (SD) of the amount of stabs in air during operation 1 and 2.

5.2.3 Discussion of the quantitative experiment results

First of all, before starting the discussion, operation 3 will not so much be in the focus when discussing the results in the section above. The operation has no tool stabbing tasks in it, and the execution with and without AR was a bit different in the way information was collected and used. With this difference the non-AR execution should by no means be completed faster than the execution with AR, this is also acknowledged from the graph in Fig. 5.12. The discussion of this operation will be more relevant when looking at the AR components included in the operation. When this is discussed later on in this chapter, the time parameter will have a partaking role.

Time spent

When looking at the time spent on the different operations, operation 1 and 2 could be looked at more or less together as they both are precision based operations that rely on depth information. Operation 3, on the other hand, is more of a gather information and make an informed decision task. First of all, it is attempting to look away from the results of participant 7 in operation 1, because of the alternative method of completing the operation. It is, however, one of the intentions with the AR system to support the user when making decisions and thus prevent such misunderstandings. It should be obvious to the operator what the operation is and how to execute it while using the AR system. This will, therefore, be further discussed when looking at the results from the evaluation of AR components. As far as for the time used it is noted as a deviation from the rest of the results.

As mentioned earlier, the participants were not expected to hurry when executing the operations, even though the experiment was timed. However, just the feeling of being timed could have caused some participants to prioritize speed over accuracy. This should though, be the same for both instances (with and without AR) so the effect of this aspect should nullify each other when looking at the time used. The reason for taking the time of the participants was mainly to have some quantitative data to measure the efficiency with. This is also relevant because efficiency is one of the usability goals used to evaluate the usability of the system. From these timed results, it is tempting to conclude that the AR system did indeed make for more efficient execution of the operations. This is solely because of the reduction in operation time for almost every participant. But before taking such a conclusion, it is not to hide under a chair that participants must have gotten a feeling for the depth of field in which the operations are executed in. Just by knowing that you have to move the tool a certain amount of distance in the depth direction (based on previous experience), can subconsciously lead to doing it faster the second time when AR is used. This builds on the possible bias discussed earlier in section 4.8.

Number of stabs

When taking the number of stabs into account, the results are even more in favor of the AR system. This is maybe a more logical one. Say if you remove one of the senses from a human. In this case depth perception. Research argue that others senses then improve or you use more brain capacity to utilize the remaining to find other sources to collect and interpret information from [119, 120]. As expected, the participants tried to use the tool to feel and check if they were deep enough into the scene to actually perform the tasks. Exactly this adaptive maneuver when not having depth information is tried captured when counting the number of stabs. When these results provide such distinct differences in behavior, it gives the indication that the AR system does provide information which to some degree generates depth perception, and thus displays both depth perception and decisions support capabilities of the AR operator interface. This is because the sum of these capabilities potentially gives the operator knowledge to support the decision of when to stab to the left in an attempt to stab the target. Resulting in a drastic reduction in the number of stabs in empty air when using the AR system.

By looking at both time and the number of stabs together, some time must have been used to stop the movement of the tool towards the target to do the horizontal stabs. It does not necessarily need to be much time, but the amount of time spent would accumulate during the operations. If the stabs indicate that time is spent on gather information through the available senses when depth is not available (non-AR), the execution of the operations with AR should then reduce or eliminate this time "waste". This is supported by the fact that the number of stabs is significantly reduced and in some cases totally eliminated, in the cases where operation time is reduced. Before stating that based on this, some time must have been saved during the AR run, other time-consuming behaviors could have been introduced with the AR system. These could be time spent on understanding the presented information and time spent on deciding what depth tool to use. There are, however, few indications towards this kind of time usage within the quantitative results, and thus the presented figures and table makes a solid case for increased effectiveness and efficiency while using the AR system compared to the non-AR system. While this is the impression given by the quantitative results, they alone are not enough to defend statements stating that the AR system reduces operation time and increase the accuracy of operations (increase efficiency and effectiveness) by a substantial margin. It is more complex than so, the need for information through qualitative results is needed to better defend any findings related to the usability of the AR system.

5.3 Qualitative experiment results and discussion

A lot of data is collected when eight test participants (section 4.2) answers a more than 90 questions long survey. Through this survey, the quantitative results are collected. These describe what the participants felt, thought and said during the experiment. To extract meaningful information from these data, like for the quantitative results, the Google Forms [104] responses were downloaded as a Google Sheet [121] and then converted to a Microsoft Excel file [118]. Within MS Excel the responses are collected in different graphs displaying different results which will be presented in this section. Because of the magnitude of the Google Forms survey, the qualitative results will be presented within one area followed by a discussion of the results within this area. There are in all three main areas, these are; subjective evaluation of the AR system, usability evaluation of the AR system and SUS evaluation. The latter is also a usability evaluation method, but it is presented in its own section to highlight the seven chosen usability goals in the usability evaluation section.

5.3.1 Results from the subjective evaluation of the AR system

In the survey completed by every participant, there is a section to evaluate the performance of every component in the AR operator interface. These are evaluated within the seven (some only five) focus areas presented in section 4.6.2. To make for a less chaotic presentation and discussion the different components are segmented into two groups. First, the AR depth tools results will be presented, followed by the operation specific AR components results, and then comes the discussion of these results.

5.3.1.1 The AR depth tools

The results from the evaluation of the developed AR depth tools (section 3.3.2) is presented in Fig. 5.15. This figure compares the depth tools to each other based on the seven factors presented in section 4.6.2. These results are based on the test participants ratings of each of the factors through the experiment evaluation (section 4.6). With eight participants answering the evaluation, the answers for each factor is averaged and then presented in the graph. The evaluation scale goes from 1 to 5, where 1 represents no cohesiveness between the evaluation factor and the component, while 5 indicates great coherence between the two. Further, the average score when looking at all seven factors together for each component are listed in Table 5.2, together with the related variance and standard deviation.

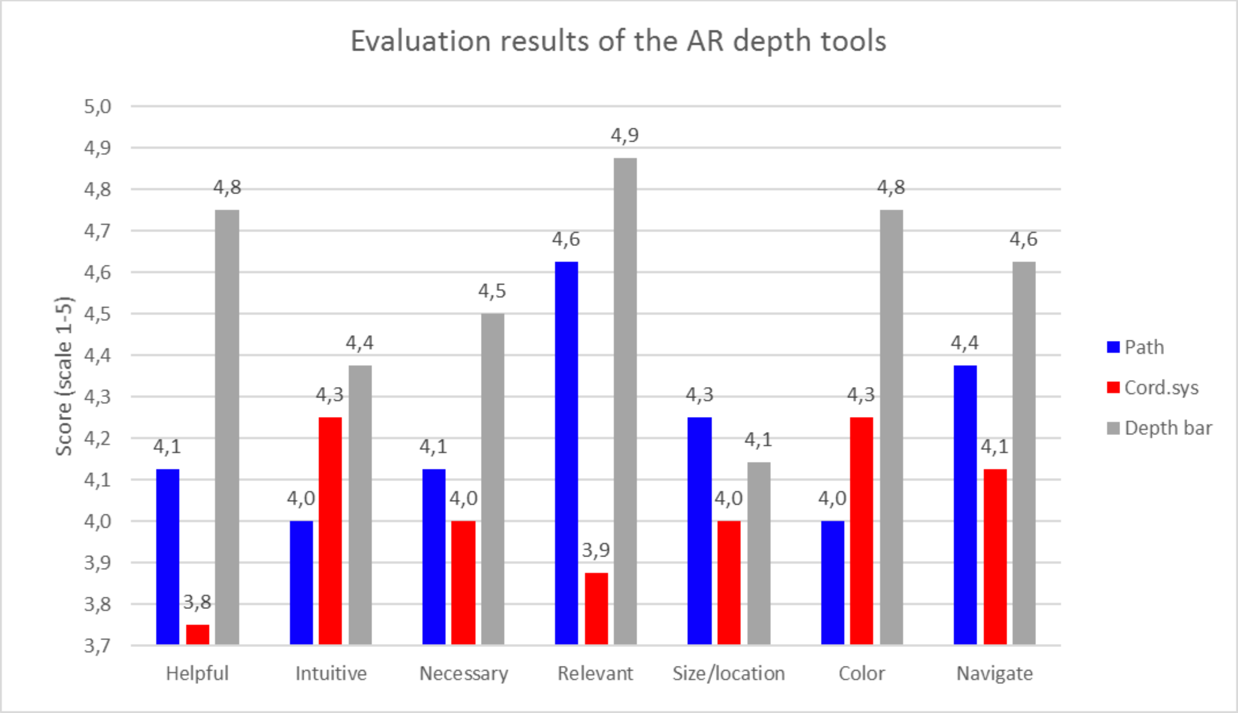


Figure 5.15: The subjective evaluation results of each of the AR depth tools.

By looking at the data displayed in Fig. 5.15, we can see that the depth bar stands out as the one component scoring the highest score for most of the different factors. This is also confirmed when comparing the average scores listed in Table 5.2. From both the figure and table it is visible how dominant the depth bar is over the two other components, outscoring the two in every factor but the size/location one. It is also worth noticing how low the coordinate system scores in helpfulness and relevance even though it presents depth information on a 2D screen. At the same time does the path score the most even throughout the seven evaluation factors, with a variance of only 0.03 and a standard deviation of 0.17. This indicates that the path might be the most reliable of the three depth tools.

Component	**Avg**	**Var**	**SD**
Path	4.2	0.04	0.21
Cord. sys.	4.0	0.03	0.17
Depth bar	4.6	0.06	0.23

Table 5.2: The average (avg), variance (var) and standard deviation (SD) between all of the seven factors of the individual components.

5.3.1.2 The operation specific AR information

The AR system is not limited to only AR depth tools. The other remaining AR components that complete the whole AR system is evaluated through the five first evaluation factors presented in section 4.6.2. The evaluation results are shown in the graph in Fig. 5.16. In this graph, the different AR components specific for operations 1, 2 and 3 are evaluated versus each other, plus the general UI (without the depth bar). These resulting components are displayed in the figures found in section 5.1.2.

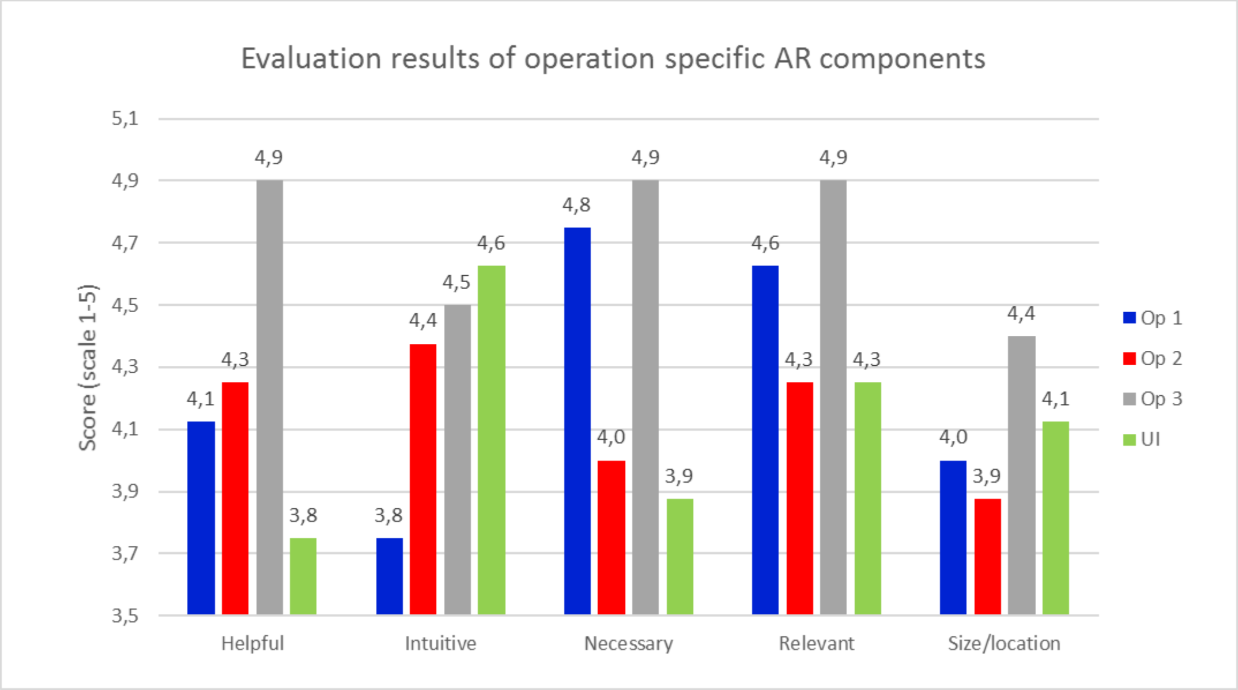


Figure 5.16: The subjective evaluation results of each of the operation specific AR components and the general UI.

Like the depth bar above, the AR components in operation 3 proves to fit nicely into its respective operation, scoring, by some margin, clearly, the highest average with 4.7, see Table 5.3. At the same time, the variance is listed to 0.05 meaning that these components scores high evenly among the five factors. Other than that, the other three components are changing between themselves on who's closest to operation 3's score for each of the factors. This results in a very even average ranging from 4.1 to 4.3 between the three. There is though more difference in standard deviation and thus variance between them, with operation 1 having the greatest spread in scores over the five factors and operation 2 having the contrary. All in all, it is the general UI that scores the lowest in this evaluation with an average of 4.1.

Eval. object	**Avg**	**Var**	**SD**
Op 1	4.3	0.14	0.38
Op 2	4.2	0.03	0.18
Op 3	4.7	0.05	0.22
UI	4.1	0.09	0.31

Table 5.3: The average (avg), variance (var) and standard deviation (SD) between the five factors of the individual evaluation objects.

5.3.2 Discussion based on the subjective evaluation of the AR system

In this section, the results in the section above will be discussed. First out is the AR depth tools which will be discussed based on the seven factors in Fig. 5.15 and then the operation specific AR components based on the five factors in 5.16. Each of them will not be discussed down to the specific focus point but rather up against each other with a more general functionality perspective in mind, to rate them up against each other and potentially discover potentials and weaknesses of the components. After the discussion of the depth tools is summarized, the remaining AR components are discussed.

The depth tools
Even though the path is placed ideally right in the action area of the participant's vision, the depth bar seems to be the component the participants were most satisfied with, when looking at the results in Fig. 5.15. It scores really strongly over the board. They seemed to find it very helpful to have a bar indicating how far they have moved, and how far they still need to move in the depth field to reach the target. With an average of 4.6, it really does outperform the two others. Still, these two also score relatively high scores, both on the upper half of the scale. An interesting point is in fact that the depth bar scores the highest even on the navigation focus point. Participants actually felt that the one-dimensional depth bar made it even easier to navigate in a 3D environment compared to the other two depth tools which provided 3D information. Such a result indicates that the two additional dimensions of navigation information could be redundant. Even so, if it is convenient to have, the participants seemed to have sufficient information from the video feed to understand how to maneuver in the right/left and up/down directions. In other words, the addition of the depth information provided by the depth bar seemed to be sufficient information to navigate the tool in 3D. On the other hand, these operations (or 1 and 2) were basically "Move forward this much until you reach your target, and do something there". It wasn't ant hinders along the way to challenge the participant to figure out a route towards the target. If this was the case, it is possible that the path would have outscored the depth bar if the path managed to propose a route to dodge the obstacles. This developed path does not have the capabilities to do this with physical objects, however, with artificial objects created in Unity such a test environment could potentially be created. This could be a suggestion for further work, to produce a more advanced experiment operation scene with artificial hinders to further test the navigational potential of the developed path.

Moving down to the opposite side of the spectrum, we find the coordinate system which scores the lowest within most of the factors and is thus the overall lowest scoring depth tool. It scores 0.6 lower than the depth bar on average, even though it presents just the same information, but through numbers and in two additional dimensions. This comes as a surprise as humans are taught to interpret distances through numeric units already from kindergarten or early in primary school. Through the results though, it is reasonable to think that the participants experienced more benefit from having the navigational information presented graphically than through numbers. This thought is also increased when looking at the responses to the survey question "What depth of field information did you use when maneuvering the tool? (Check the ones you used)", the responses are shown in Fig. 5.17. Not surprisingly, these results support the ratings presented in Fig. 5.15. Six out of eight used the depth bar during the experiment and only two did use the coordinate system. The path scores in the middle in both of these graphs. Nevertheless, with only two participants using the coordinate system, it raises the question of "did the remaining participants even notice how the coordinate system functioned?" If they did not really focus much on it, the subjective evaluation results could be influenced by this and could be a potential explanation to why it scores the lowest. Could it be that the discussed in section 5.2.3 reduction in operation time with the AR system came at the expense of participants not taking their time to truly look at the functionality of every depth tool? Seen from another view, was there maybe other depth tools that were more appealing to the participants and thus they were chosen to be used over the coordinate system? Or was it down to the placement in the operator interface like discussed in section 5.1.3? Because, from a developer point of view, the depth bar has a huge limitation in that, if the tool is moved past the target, the depth bar simply indicates that the tool is at the target, and gives no indication about the fact that the tool has passed it. This is where the coordinate system simply puts a "minus" in front of the distance number, indicating that the tool has to be maneuvered a certain distance in the negative direction to compensate. The path as well, the lines switches from going into the scene to towards the participant when the target is passed. When going through the screen recordings form the experiment, only one of the eight participants moves the tool past the target, thus only one out of eight is exposed to this limitation of the depth bar. So, even though it is not reflected in Fig. 5.15, it should be remembered as a sizable limitation.

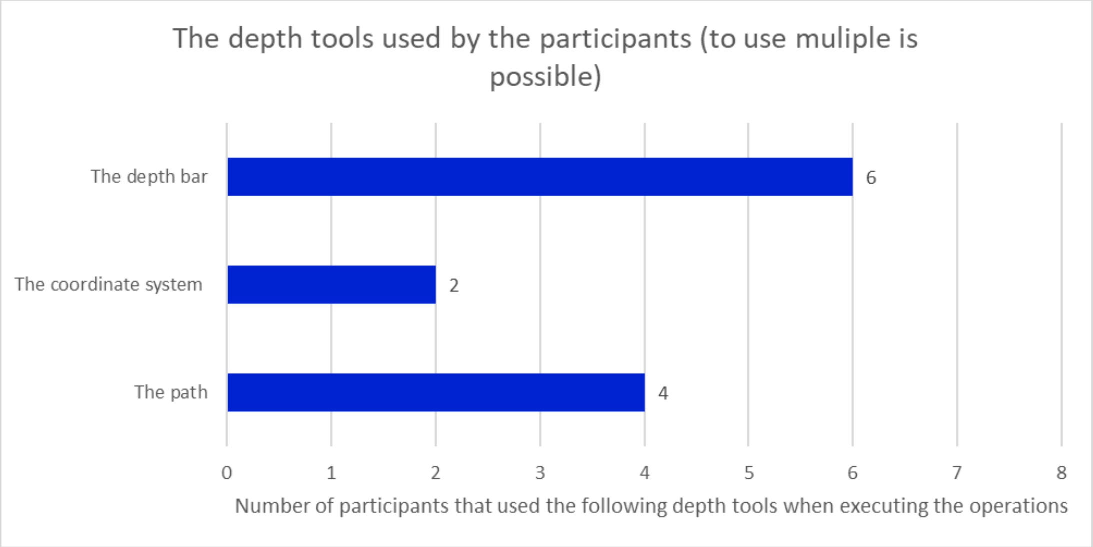


Figure 5.17: Results when asking the participants to name the depth tools they used when executing the operations.

When deciding to go with three different depth tools to get individual feedback on them, a drawback is that it is not given that each of the participants uses everyone one of them. If that is the case, they are not fully capable of giving their actual feedback on every component. But, every participant was informed about the three tools and got time to get familiar with them, so hopefully, they have seen each of them, looked at how they work, and then made an opinion of which they wanted to rely on when executing the operations.

Despite the limitation of the depth bar, it is understandable that participants like to use it in their search for depth information. It provides just what's needed, information about how far you must move to reach the target. It even does it in a very intuitive manner. With maybe improved placement in the scene and a solution to the negative depth limitation, the depth bar, based on this subjective evaluation, emerges as the best candidate between the depth tools to provide depth perception capabilities in an AR operator interface. The potential of the path should at the same time not be forgotten. It could be a very powerful tool to have a path going from your position to the target, and this presented right at the visions focus point. But for this to work, more weight has to be placed on more seamless integration of the AR path into the scene and development of an adaptive path that can account for physical hinders in the pathway. It would be interesting to look at the possibilities of utilizing big scale adaptive path planning concepts and ideas such as [122, 123, 124] to use as a first-person view AR path. If this ever becomes realizable, it would come a long way in increasing collision safety during operations and then probably drastically outperform the depth bar in later evaluations.

To summarize the subjective evaluation of the depth tools, there should be no doubt that these are both relevant and necessary tools in this context. With 19 out of the 21 bars in Fig. 5.15 scoring 4 or better, it seems like the participants, overall, is satisfied with the general performance of the AR depth tools. Nevertheless, solely based on these it is the coordinate system who draws the weakest straw. At the same time does the depth bar emerge as the component to bet on, while the path doesn't really seem to fulfill its potential, due to implementation difficulties. It is, therefore, based on these results and discussion, recommended to continue the development of the AR interface with the depth bar as the depth tool if it is freed from its limitations. The path should at the same time not be forgotten, as its potential capabilities of providing depth and navigational information far proceed the depth bar's potential to further evolve and answer the path's potential.

A general remark on size and location

By looking at the evaluation results of the AR components in every group, the size and location scores are both in Fig. 5.15 and 5.16 amongst the lowest scoring focus areas. A reason for this might be that every component combined together overcrowded the scene a bit, and thereby collectively drag each other down. Thus the participants might have felt overwhelmed and did not rate the size and location of any component as perfect. It would have been interesting to try to increase the field of view of the workspace within unity, i.e., the potential AR implementation area of the video captured. This to reduce the size impact of all the components collectively to not hinder so much of the visibility of the surroundings. Because while using the ARCamera from Vuforia in Unity it produces a narrower field of view than when simply running the web camera in the camera application on the computer. So if the field of view was increased or the system was displayed on a bigger monitor, probably the participants would have rated the size and location better. On another note, the idea of performing the same evaluation with only one depth tool is compelling, to see if or how responses in this category would change. This is interesting for future work if one of the depth tools is recommended as the best usable component to provide depth perception capabilities.

AR components

For the AR components other than the depth tools, Fig. 5.16 shows the subjective evaluation results. When comparing these to the similar results from the depth tools, one realizes that there are 2 fewer evaluation factors. Navigation and color are removed, because first of all, these AR components are not designed to give explicit navigational information like the depth tools and secondly the colors are chosen to fit the operation scenes in the experiment and not so much any arbitrary scene. Therefore, it felt superficial to ask the participants to evaluate these factors. Moving forward, the operations and their respective AR components and the general UI will be discussed in the next paragraphs.

Operation 1

Starting out with operation 1 who scores, on average, second highest amongst the four tested component categories with an average of 4.3. There is though, quite some variance in the results ranging all the way from a necessity score of 4,8 down to the low intuitive score of 3,8. When looking at the AR effects in this operation, see Fig. 5.4, it is in a way understandable with such a variance in the evaluation responses. Surely, it is helpful to see the pattern of a hidden maze you are about to complete, but could there have been added additional information to make it more intuitive for the participant to understand how to interpret the information and how to follow this pattern to complete the maze? Probably yes, looking back on the participant's execution of operation 1, the AR system should've made it more obvious to the participant were to enter the maze. This could for instance been indicated by an artificial arrow pointing at the intended entrance. It was thought during the development that the path together with the operation handout, see appendix A, were sufficient information, it turned out, however, to be the biggest uncertainty for the participants during this operation. This uncertainty leads to the one instance where a participant did not manage to properly complete a task within one of the operations. In this case, the maze was "cheated" by simply stabbed straight through the center hole without "solving" the maze. This instance was mentioned in section 5.2.1 and it further emphasizes the need for an entrance indicator. In addition to such an indicator, a method to relate the physical maze to the AR maze so it is intuitive to understand how the AR maze is orientated in relation to the actual maze could also be beneficial. Because, how it appears now, it is not trivial to the user where the entrance of the AR maze is found on the physical maze. The actual entrance area spoken of is the one highlighted in Fig. 4.12. Such additions to the AR components in operation 1 seems needed based on the presented results from the subjective evaluation. Especially because the camera tilt discussed earlier and other discussed factors, could have impacted the alignment of the path, which then would not as clearly as first intended illustrate the maze entrance.

The top scoring factors in operation 1 are necessity and relevance, indicating that the participants thought there definitely was a potential for supportive information to help execute this operation. But due to visually lower scores on the three other factors, which highlights more the actual information presented and not so much the concept of having this information presented like the two others, it is reasonable to think that there is room for improvement in how the AR components were presented in this operation.

Operation 2

For operation 2 we see an average score of 4.2, which is right behind operation 1, however, operation 2 has significantly lower variance (information from Table 5.3). It scores higher in two of the areas where the operation 1 AR scores relatively low. These are helpfulness and intuitiveness. Maybe this is because the AR objects in operation 2 had what operation 1 might have been lacking? Namely, an indicator to show where to enter the quadratic hook. But in this operation, where to enter the hook was probably way more intuitive, since the constraints on the tool only let it stab to the left, then the only way of entering the hook was by approaching it from the right side. This is also why the indicator is shifted slightly to the right of the hook, to indicate what side to enter from. So, the need for such a big indicator when it is kind of obvious where to enter from can be discussed. It is, however, included to make it fully clear for the participant what to do, because at the same time it works as supportive tool in the way of saying "you remove the dirt by entering here" which might be needed due to the maybe not so obvious link between dragging a piece of paper off the wall to clean the equipment. It also works as additional support to help avoid misunderstanding such as the one instance during operation 1. When speaking of support, the appearing text confirming the cleanness of the marker to the participant seems to be helpful, and it must also be reassuring to receive such a confirmation when you are supposed to tell someone with confidence that something is done properly. On the other hand, must this information be presented in a convenient manner, not unnecessarily blocking line of vision which could reduce situation awareness or overwhelm the user through too large information boxes. This might have, to some extent, been the case in this operation. Since, the participants evaluate the size and location to 3.9, which is operation 2's lowest score and the lowest in that focus category.

Experience to extract from this operation is then probably to better differentiate between relevant information and necessary information. If the purpose of the indicator was to explain the link between the hook and cleaning of the equipment, which it was, this could have been solved differently. Ideally though an AR decision support component less area expensive. Maybe this information could appear in the general UI? Maybe the clean confirmation also? The depth tools on their own might be enough AR information in this operation and if this were the case, more of the participant's focus could then be targeted towards these tools, especially the path which guides the participant directly to the correct entrance of the hook.

Operation 3

Like the depth bar stands out amongst the depth tools, the results for operation 3 also stands out in Fig. 5.16 with a very high average score of 4.7. How this operation is executed is through a bit different between the non-AR and the AR run. In the two previous operations, the participants have done just about the same when executing the operations with the two different systems. In this operation, the execution becomes a bit different depending on the system. The AR system has the operational information, else found on different monitors when operating an actual subsea ROV, see Fig. 5.18, integrated directly into the system. On these monitors, all information displayed might not always be filtered to show just the most current operation information from time to time. In the AR interface, it is intended to let the system filter out the excess information and only present the information most relevant for the operator in order to have relevant information available when important decisions have to be made. This is in an attempt to utilize the powerful functionality of AR that it can help the operator navigate through general information density by having adaptive information visualizations within the operator interface [125].

Figure 5.18: The operating room of multiple subsea ROVs, illustrates how the operators have different screens with information to backup the main video feed from the ROV. Source [126].

For the non-AR system run, the participants received unfiltered operation information on some sheets of paper to use when solving operation 3. From both the time spent, presented in Fig. 5.12, and the results from the subjective evaluation in Fig. 5.16, it becomes quite clear that participants benefited from the use of the AR system. Sure, the operation time will be reduced when all relevant information is presented to you in the system while being guided through a button sequence where you make your decisions, see Fig. 5.6 for the actual execution. But seeing as the participants also rated the AR components very highly in the subjective evaluation indicates that they enjoyed the convenience of having everything available in the system. This solution seems to score very highly in helpfulness, necessity, and relevance, meaning that the potential of supporting and improving the work of real subsea ROV operators is most probably there. This would though require to add their real IMR operational information into a similar system.

For the AR implementations used in this operation, the participants used some time to understand that the operation list now was displayed on the screen. This operation list was previously, during the non-AR run, given as a handout, see appendix A. They also used some time to understand that they should use the button sequence to solve the operation. If this is due to the participants not expecting this information to be integrated into the system, or because of the size and the placement of this information, or due to lack of intuitiveness in the presentation, is uncertain. It could also be a combination of these three factors and even others not mentioned. What is certain is that even though operation 3 scores high in both intuitiveness and size/location, they are both significantly lower than the other three factors, indicating that this actually is the weakness of the operation specific AR components.

The discussion above is interesting, as a system who holds the solution to a problem is rather useless in supporting the user to solve the problem if it is not easy or intuitive for the users to understand how to properly use its functionality [42]. With this in mind, improvements to make the AR components more inviting to use could be beneficial. The best scenario would be if the user automatically was dragged towards the relevant information in a given situation. This is kind of what happens after the participant's first couple of clicks in the button sequence. After these clicks, their focus is locked, because they expect to find the next piece of follow up information in that same area. At this point, it is pretty easy to drag the user towards the next relevant piece of information because they expect to find it within the same area. It would of course not be wise to place a followup dialog box at a completely different place in the layout. So, the challenge here is really to capture the participant's attention at the right time and guide their focus through the area with relevant information. Maybe this could be done by adding some effects to the button sequence? If the dialog box blinked for a period of time when it appears as a valve is detected, could that help draw the participants instant attention? In addition, as the first button is clicked, the focus has to be shifted towards the operation lists to decide if any of them indicate another valve level than the current, maybe then the operation list boxes themselves could blink? To make it clear for the participant that this piece of information needs their attention. These suggested improvements could potentially make it more obvious to the participant how to navigate their focus between the different AR components. This would hopefully make it more intuitive to use and give more reason to the location of the information. At the same time, it must not be forgotten to not overcrowd the scene with AR objects and blinking effects which could overwhelm the user, it must be integrated in a thought through manner.

General UI

Finally, while looking at the general UI (i.e., the green bars around the image), it is the lowest scoring of the AR components. It serves, however, not the immediate purpose of helping the participant out in completing the operations. It is there more to provide general information and add the functionality needed if deviations from standard procedure occur. It could be looked upon as the component who is a part of the equation of which the sum is the felt decision support, from the system, by the participant. Another reason for it being a part of the system is to satisfy the minimum requirements of NORSOK U-102 [106]. This is to make the link towards use in actual subsea IMR operations more realistic, as such standards must be satisfied almost even before thinking of implementing it. As discussed in section

5.1.3, the UI serves the purpose to standardize where to expect to find certain information. An example of this was the placement of the main menu button which the participants had to click after every AR operation. The placement of this button was showed before operation 1 started. Almost everyone had to ask again where it was after operation 1, but after operation 2 only a few asked again and after operation 3 no one asked. This is exactly the kind of thing this general UI seeks to achieve. In order for this to work in a broader setting, it has to be presented in an intuitive way. If the information is not presented so that the participant understands it from the beginning and it just keeps appearing all the time when using the system, it would just build frustration rather than serve the purpose of supporting the participant. It is, therefore, a great foundation that the UI scores high on intuitiveness in the subjective evaluation. Overall though, it is the lowest rated component in Fig. 5.16 with an average of 4.1. But again, this is not the component providing the most help during operations, it might not seem so necessary, and it does consume significant space on the screen. Even so, it serves a purpose broader than what is tested in this experiment. It is aimed more towards corporate use and to provide familiarity, utilities, and flexibility during a variety of missions.

From the results in section 5.3.1 and the following discussion in this section we have seen how the different operation specific AR components have its own strong points and weaknesses. Same as for the depth tools, these components scores rather high overall. Thus, the participants must have felt some kind of support from the AR system while completing the operations. Even though some AR operation components seemed to utilize the decision support capabilities in a better way than others. Operation 1 stands out as the one with some missed potential while operation 3 seems to manage this the best. The concept of the latter is also something to focus on in further development, also because this experiment operation can rather easily be related to an actual subsea ROV operation as discussed in section 4.8. The value of moving already collected information, found in databases, to the operator interface so the operators don't have to move their focus to another monitor to check or find information has been proven by these results to be substantial. Which is why this operation stands out as the operation which provides the most support to the participants when important decisions with high stakes are to be taken.

5.3.3 Results from the usability evaluation of the AR system

Having evaluated every aspect of the developed AR operator interface in regards to different evaluation factors related to their performance, it is now time to shift the view to the whole AR system. With the focus now changing from component level to the whole system, results are in this section is to be presented based on the usability evaluation described in section 4.6. In this evaluation, the AR system has been evaluated both as a single performing system and compared to the regular video stream system (non-AR system). With the usability evaluation, more emphasis is placed on discovering limitations in the system that hinders the operator in utilizing the depth perception and decision support capabilities and also on ensuring a great user experience. The results in this section are based on responses to relevant questions and statements within the different usability goals. The responses in each of the seven groups are converted, summed and averaged to fit the scale.

5.3.3.1 The AR system

Before we present the results it should be noted that, when evaluating the usability of the AR system, every participant was excited about the concept and purpose of the system. However, in this case, it is only the actual functionality of the developed AR system at hand that is evaluated, not the idea or concept of having a fully functioning decision support and depth perception system.

In Fig. 5.19 the graph displays the resulting usability scores on a scale from 1 to 5. On this scale, 1 represents worst usability (i.e., the system is working against the user) and 5 represents a system that is greatly helping the user along in achieving goals within the usability area. With this in mind, the system is scoring overall on the higher part of the scale with no usability goal scoring lower than workload's 3.5. In the other end of the spectrum is learnability and satisfaction with their tied score of 4.5. This means that all seven goals are scoring within a range of only 1 on the 1-5 long scale. Naturally, this results in a low variance, which is also confirmed by the variance value of 0.14 listed in Table 5.4.

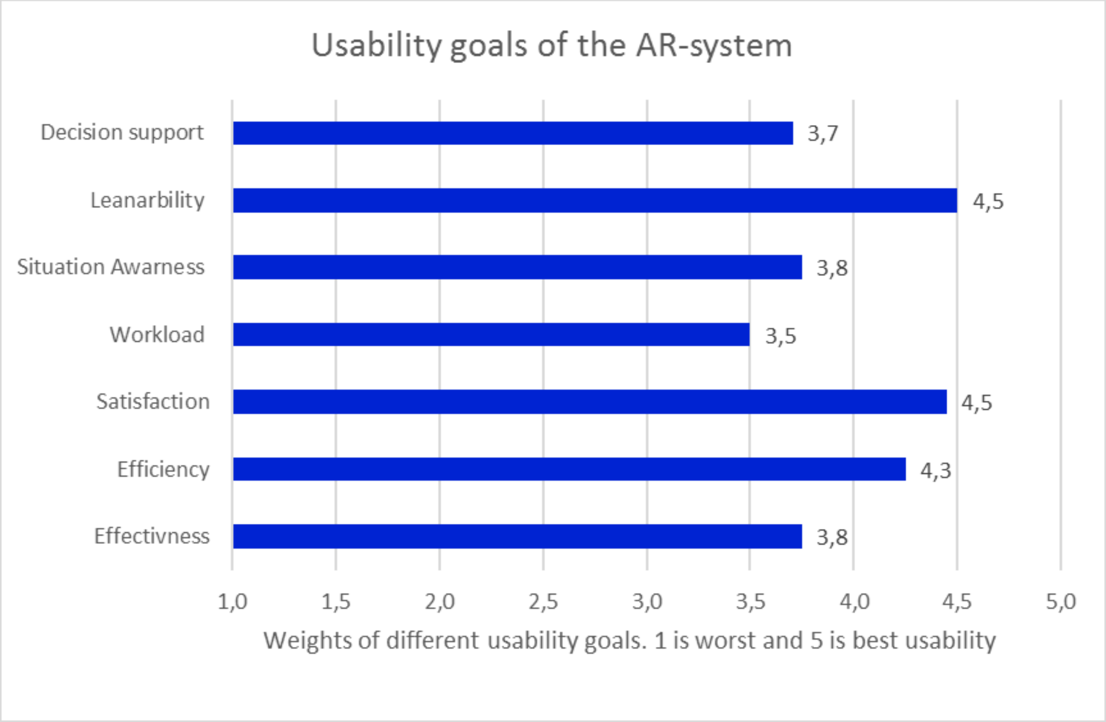


Figure 5.19: The usability evaluation of the AR system results.

In terms of the presented scale, the average combined score of 4.0 from Table 5.4 indicates that the developed AR system provides good, but not great, usability when executing the operations in the experiment (section 4.4). It is also worth noting that the users of the AR system seem to think, based on the results, it lets for more efficient operation execution through an easy to learn, understand, and get going with system in which they are satisfied with the overall performance of. Meanwhile, it appears that the system did not rate as high when it comes to supporting the users, providing a sufficient level of workload, and giving them great awareness to execute operations in an accurate and complete way. It is worth mentioning that both the amount of stabs and time used from the quantitative results is not weighted and added to this data. The presented results are, therefore, solely based on the subjective responses to the usability goal targeted questions and statements.

AR system	**Avg**	**Var**	**SD**
Usability	4.0	0.14	0.37

Table 5.4: The average (avg), variance (var) and standard deviation (SD) of the usability goals together.

An important aspect of the developed system is the ability to aid the users during operations. Therefore, this ability is highlighted in Fig. 5.20 which displays the survey results of the level of support the users felt from using the system in each of the operations. Clearly, the users were satisfied with the support during operation 3, as they unanimously agreed that the system offered great support for that specific operation. The story is not so much the same for operation 1 and 2, with the first scoring a low of 2.5 and the latter having a decent score of 3.75.

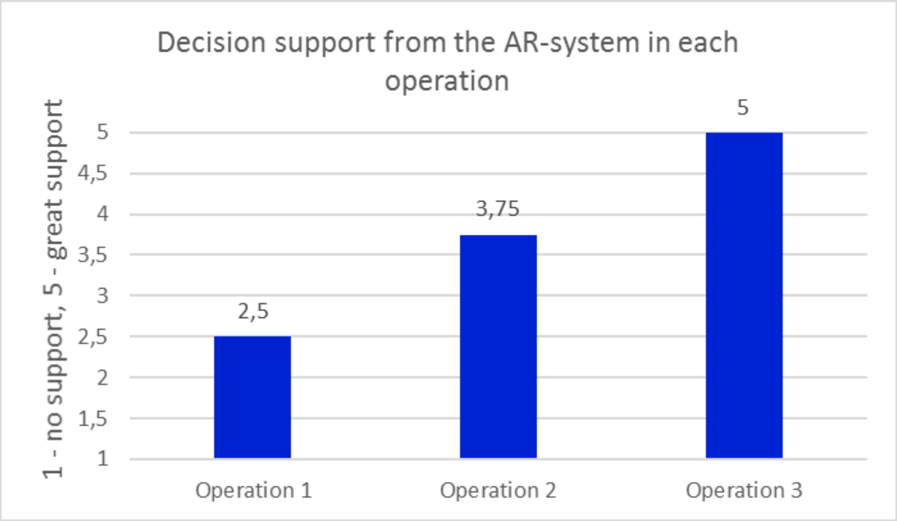


Figure 5.20: The results from evaluating the decision support offered by the system in each of the three operations.

5.3.3.2 The AR system compared to the non-AR system

After looking at the general usability of the developed AR system, this same system is compared against the non-AR system consisting of only the pure video feed from the camera (section 4.3.2). The data gathered from the survey described in section (4.6.1) is presented in Fig. 5.21 and the corresponding Table 5.5.

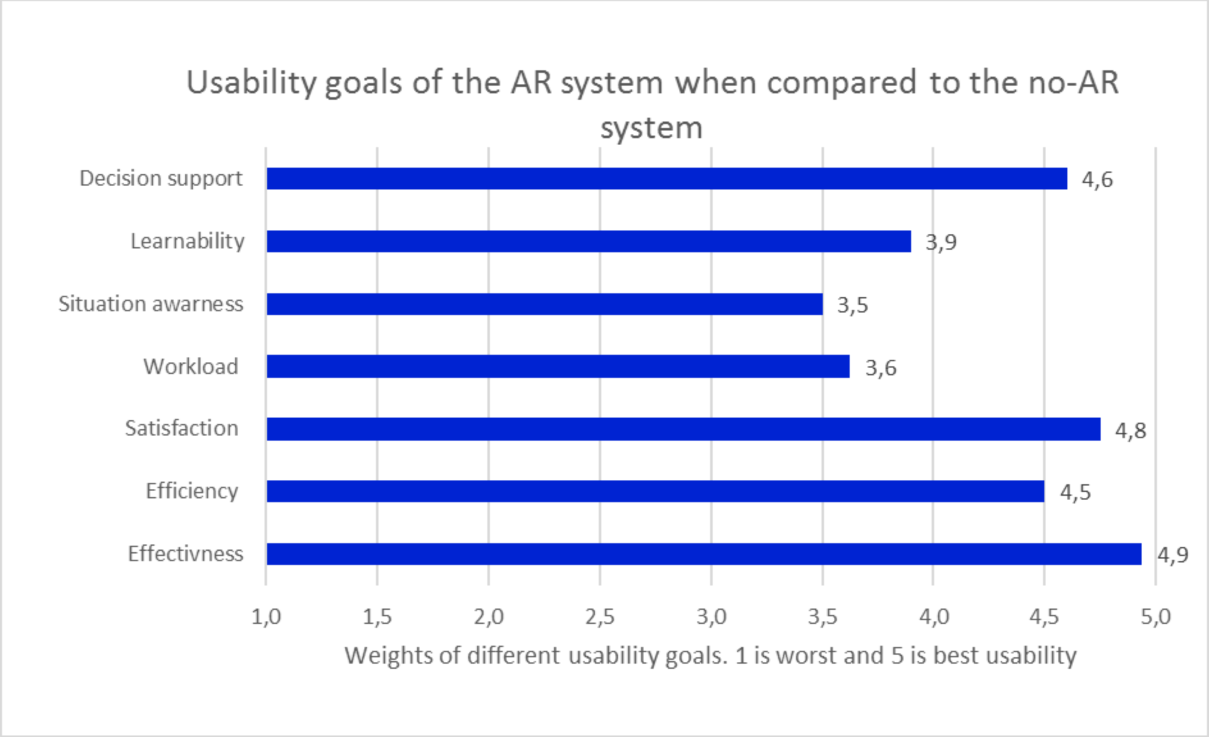


Figure 5.21: The evaluation results when evaluating the usability goals of the AR system compared to the non-AR system.

The immediate impression when looking at the graph in the figure is the split between the high scoring goals and the lower scoring ones. This is also reflected in the standard deviation which is 0.53 followed by a variance of 0.28. Compared to the non-AR system it seems like the users enjoyed the experience using the system with the support it provided and the resources which made way for more efficient execution of the operations. At the same time, it appears that the AR system did not reduce the workload nor increase the situation awareness to the degree of the other higher scoring usability goals compared to the video feed of the non-AR system. Even so, the AR system is overall scoring high with an average of 4.3, showing that the participants, by a huge margin, preferred the AR system over the non-AR system. If the participants rated the non-AR system higher than the AR one, the score would have been much closer to 1.

These same impressions are generated when looking at the results from a “check the fitting statements” question in the survey. From the results in Fig. 5.22 one can see that 50% of the users thought the AR system provided faster understanding of the operations, while everyone thought the system provided support and better access to relevant information.

AR vs non-AR	**Avg**	**Var**	**SD**
Usability	4.3	0.28	0.53

Table 5.5: The average (avg), variance (var) and standard deviation (SD) of the usability goals together compared to the non-AR system.

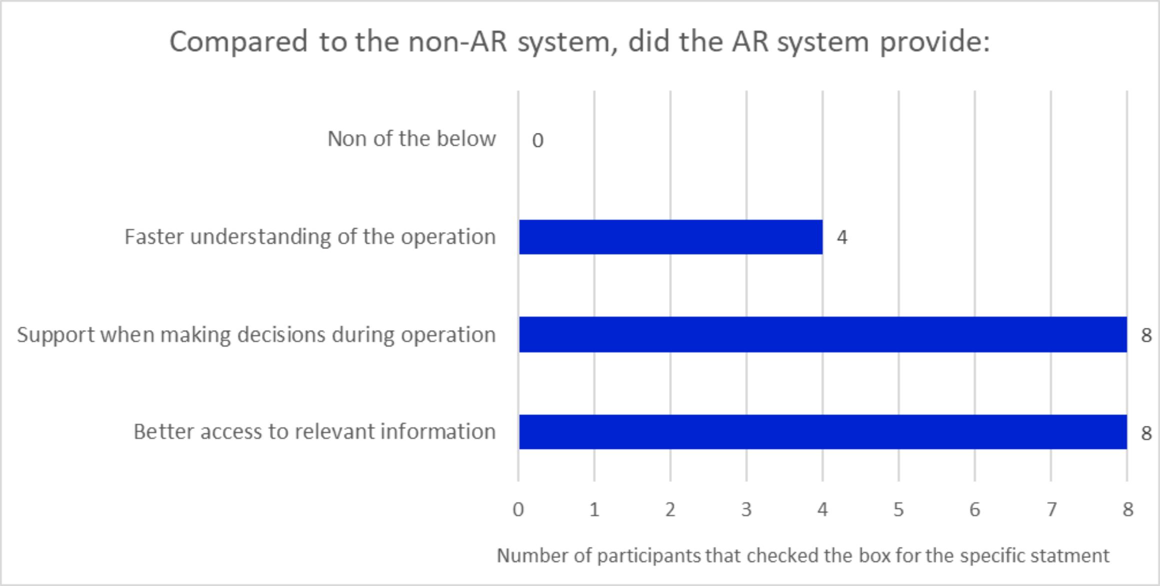


Figure 5.22: The evaluation results of each of the AR components versus each other.

Finally, Fig. 5.23 provides a comparison between the score of each usability goal when looking at the AR system alone (the blue lines, bars from Fig. 5.19) and the results when comparing the AR and non-AR systems (red lines, bars from Fig. 5.21). The goals standing out is definitely decision support, learnability, and effectiveness. These have a difference between the two evaluated scenarios of 0.9, 0.6 and 1.1 receptively. The remaining four goals are more equally rated in these scenarios with workload having the lowest difference of 0.1 and satisfaction having the greatest with 0.3, making the score difference between these four goals ranging from only 0.1 to 0.3. Thus meaning they are evaluated rater similarly by the participants in the two different scenarios.

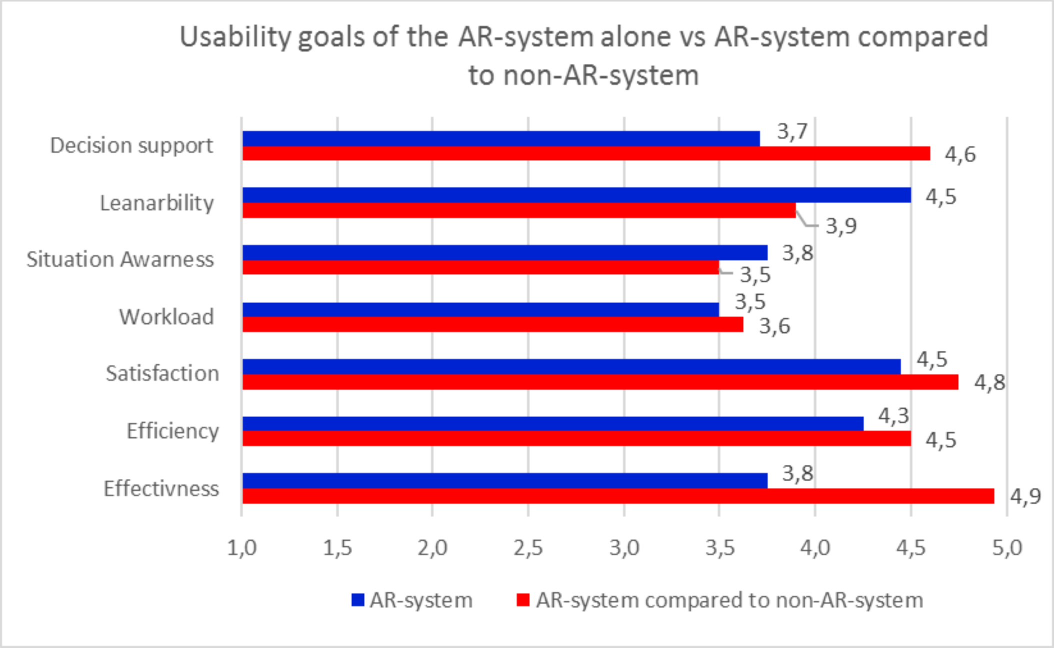


Figure 5.23: The usability goals scores from the evaluation of the AR system alone vs the benchmarking scores. The scale is the same as in Fig. 5.19 and 5.21.

5.3.4 Discussion of the AR system usability evaluation results

As seen above in section 5.3.3, the AR system has been evaluated based on usability alone and usability versus the non-AR system. The results in Fig. 5.19 indicates the general usability of the stand-alone system, while Fig. 5.21 looks at how the usability of the AR system is, compares to the non-AR system. When looking at the general usability, these results are lower than the system compared usability. This is maybe not so surprising since when evaluating the system on its own, the participants should evaluate based on the standard they expect and desire when using a system. Different from when the AR system is compared to another system, then the participants should evaluate based on its usability versus the opposite system. Since the setup used while operating ROVs during subsea IMR operations is without AR like the non-AR system, this comparison kind of resembles a comparison between the used standard while operating ROVs and a potential improvement utilizing emerging technology. In this section, these usability evaluation results will be discussed to point out weaknesses and strong aspects of the usability of the AR system with depth perception and decision support capabilities, especially in mind. This will be done by looking at the usability goals one by one, these are earlier defined theoretically in section 2.3.1 and defined specifically for this experiment in section 4.6.3.

Effectiveness

When looking at the effectiveness by itself, which is sort of a performance indicator. Then it actually is one of the lowest scoring usability goals. The AR system was expected to score

high here, as one should think a system which aims to provide depth perception and decision support would enable the users to carry out the work they actually intend to do. It scores 3.8 out of 5 in Fig. 5.19, which by any means is not a bad score, but it could have been better. This say "shortcoming" might be due to the robustness of the system. Sure, the system should enable the participants to execute the operations, but was the participants left with a feeling that the operations were very constructed and thus the system only would function in this setting? If so, that could reduce their opinion of the robustness in the context of operational flexibility. Or is this result of the robustness and reliability discussion in section 5.1.3? One of the reasons for asking these questions is based on the fact that the user had to go back to the main menu and select the specific operation to be executed in order for the AR system to function properly. In addition to this, some of the AR graphics was not behaving properly during a few of the participants run. Because of this, a short restart of the system or off/on toggle of the AR component was needed. This could've of course had its influence on some of the participant's evaluation of the general performance of the system. Robustness, as mention earlier, is a great difficulty when implementing AR. Making sure that computer-generated AR objects are aligned with the physical reality, is an area which needs more focus to improve the robustness and reliability of the presented information.
Effectiveness is a usability goal where the robustness and reliability aspect is one of the most crucial areas to look at when speaking of information providing systems [127]. So, if we look at the big picture, no company is encouraged to gamble on implementing a new system as long as they are not ensured high reliability when it comes to presenting correct information at the appropriate time, in a system that is robust enough to function in a wide range of different situations. Therefore, more work has to be directed towards robustness and reliability in AR integration, before such systems as this developed can even think of being used during subsea IMR operations.

Figure 5.24: The effectiveness score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

The bright side though is the results of the effectiveness of the AR system compared to the non-AR system. It is actually the highest scoring usability goal (see Fig. 5.21 or 5.24). The reason for this might be that the participants potentially felt like they had more information about the depth field when using AR compared to not using it. With this information, the participants might have been supported to more accurately perform the stabbing operations compared to without, something that is indicated by the results in section 5.2.2. This also shows the importance of conducting both a usability evaluation based on the system on its own and a comparison between two different scenarios. The take away when comparing these results, as in Fig. 5.23 or the one presented here Fig. 5.24, could be that the participants are very happy about how the AR system helped them conduct the operations, but at the same time thinks there is potential for improvements in how the system performs at its own. A recommendation for further work is thus to emphasize time towards the integration of AR information, potentially by utilizing sensor fusion, or a markerless method which better understands the scene and provides more ownership of the system, compared to the closed-source Vuforia engine. These recommendations would hopefully help improve the robustness and reliability of the system and thus enable the depth perception and decision support capabilities to function seamlessly in their intended roles.

Efficiency

If you want to enable your users to complete tasks efficiently you want to let them do the right things in order to complete their task. In other words, you want to spend as little resources as possible to complete the task. From the results presented in the section above and summarized in Fig. 5.25, the efficiency of the AR system scores very good. It seems like the participants felt like they did not waste much time when using the AR system and the score also indicate that they were enabled to use their resources towards completing the operations. This is supported by the quantitative results through the cut in operation time and the reduction in the number of stabs in empty air which point to fewer resources spent towards miss judging the position of the tool in the depth field. The reason for this might be that the AR system is successfully in supporting with relevant information to the participants during the operations. To what extent it manages to provide this supportive information is discussed in section 5.3.2. This reasoning is motivated by the fact that every participant felt the AR system provided better access to relevant information compared to the non-AR system (see Fig. 5.25). And even though the usability of the AR system scores 0.2 lower than the compared usability, it is reasonable to also think that participants felt that the AR system in general presented relevant information to complete the operations in mind, but with some potential for improvements as discussed in the earlier mentioned section. So, it is absolutely feasible to think that spending resources on implementing depth perception and decision support capabilities in an AR operator interface are resources spent towards helping the operator completing the operations. Which means efficient expenditure of resources.

Figure 5.25: The efficiency score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

Satisfaction

An important component in making a usable system is to make it so that people actually want to use it. That is why it is important to evaluate the satisfaction of using the system when looking to evaluate the usability of the developed AR operator interface. A goal must be that no user should at any point feel discomfort while using the system.

From the usability evaluation of the AR system, the blue bar in Fig. 5.26 express a good attitude from the participants towards the system. Meanwhile, does the comparison based usability evaluation result in an even better score (the red bar). Indicating that the participants essentially had a great user experience while using the developed system. This is a good confirmation to receive, namely the feeling that the implemented AR system seems to be something the participants want to use. Which also is supported by one of the final questions asked in the survey, presented in Fig. 5.27.

Figure 5.26: The satisfaction score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

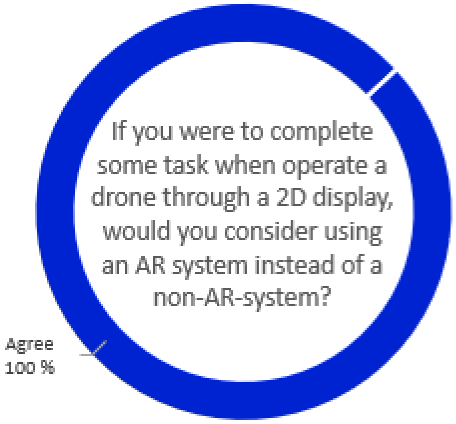


Figure 5.27: The responses to a survey question about whether or not the participants would consider use an AR system like this again.

When continuing the work to further develop this system or similar systems, it is important to understand that satisfaction scores could be reduced in later usability evaluations, if the focus is shifted completely towards improving the score or performance of other usability goals. That is because the user experience is the comprehensive impression of the whole system. So, when adding to, removing, or editing components within the AR interface the satisfaction score from previous evaluations is not so much valid anymore. Take the potential improvement by introducing blinking information boxes, suggested during previous discussions in section 5.3.2, as an example. These boxes might achieve their goal of drawing the users attention, but it is not a guarantee that this change would result in the same or even higher satisfaction scores in future evaluations. The satisfaction aspect is, therefore, something to keep in mind during the whole development phase, it is not something to forget in further work even though one usability evaluation ended with an almost great satisfaction score.

Workload

The joint lowest scoring usability goal both when evaluated as a standalone system and when compared to the non-AR system is workload. These scores are shown in Fig. 5.28. When looking at the experiment conducted in this report and presented in section 4, you could get the impression of it being a bit demanding. The participants are supposed to perform precision operations with a manual tool held up and maneuvered by their own physical strength while being timed by the practitioner. At the same time is their only visual sight of the scene they operate in presented on a small 12-inch screen. This, collectively, builds up the foundational level of workload during the experiment. It must, therefore, be the AR systems objective to not add much more load during the operations. However, with the amount of information the participants have to grasp over and remember from the 10 minutes of introduction before the operations start. In addition to all the AR information, they have to get familiar with during the different operations, is likely to add some more load and thus demand more focus from the participants. Following the importance of balancing the workload, making sure the participants are not under or overloaded during the operations [43], it might, in this case, be more relevant to make sure the participants are not overloaded. The fact that this developed AR interface also uses three different dynamic depth information tools doesn't exactly reduce the load either. So the score of 3.5 and 3.6 is probably reflected by the possibility that the participants, to some degree, felt significant levels of both physical and mental workload while using the system. In addition, even though the participants were told not to be bothered by the fact that the operations were timed by the practitioner, the low operation time by some of them, presented in section 5.2.1, could indicate that some felt like they were under time pressure and thus also some degree of temporal workload. With this in mind, focus should be directed towards reducing the load rather than increasing it when speaking of balancing the workload, as it seems like a significant degree of load was put on the participants during the completion of these operations.

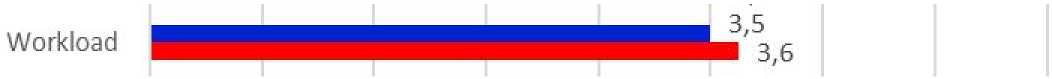


Figure 5.28: The workload score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

When users are exposed to significant load during usage, it is important to keep the vigilance aspect in mind (see section 2.3.1 - workload). Would the AR components in the AR operator interface reduce the users capacity for sustained attention to discover occurring situations, deviations, or detect critical operational signals to which they have to respond to? Does the AR depth tools require too much attention so that other critical information is overlooked? These are all interesting aspects to look at when keeping this potentially significant workload on the participants in mind. It could, therefore, be interesting to have some objective measuring method of the workload, to understand more of the, in this context, week workload results. This could, for instance, be measured by having some discrete information appear during the operation, which the participant would have to respond to. The time then used before the participant noticed this appearing information, could be a measuring point of the participant's vigilance while using the AR system. The level of vigilance would directly refer to how the workload "distracts" the participants from being great and observant operators [43]. From these results presented, it seems like, if the operations where more complex and more similar to real life operations, the participants would maybe not be able to complete all the tasks due to the workload. But in a more real-life scenario setting, the tool won't be maneuvered by physically holding it, reducing the physical demand. In addition would there ideally only be one depth tool, which hopefully reduces the mental demand of navigating through all the AR information in the interface. For further work, it would, therefore, be interesting to evaluate the workload in the more real-life scenario mentioned above, with the objective measuring point suggested. This would be interesting to further investigate in what way the system adds load, rather than reducing mental workload potentially coming from being unsure about decisions and clueless about the whereabouts in the depth of field during non-AR operations.

SA - Situation awareness

Another of the lowest scoring usability goals is situation awareness or SA for short. It is one of the two instances where the AR system evaluation scores higher than the score of the AR system relative to the non-AR system, as seen in Fig. 5.29. It is, therefore, not so unreasonable to think that the participant felt like their SA was reduced due to all the AR components in the AR system. The number of depth tools does not help the score of this usability goal either. These take space in the AR interface, and thus reduce the overall view of the surroundings compared to the empty non-AR system. At the same time should it be one of the AR systems objectives to, through AR, make the user feel in control of the surroundings, despite the fact that AR potentially produces more blocking objects in the scene. A way to do this is by making sure the AR objects contains valuable information which provides the user with knowledge to be better equipped to make a decision or take action [44]. This could be information which describes something in the environment (e.g., the maze profile in operation 1), a depth tool which gives awareness of the depth field in the captured scene, or a wide angle camera lens that provides a broader perspective and thus captures more information about the scene displayed on the screen. Based on the results it is reasonable to think that every bit of AR information does not fully fulfill this objective, or the information provided by specific AR objects might not justify its space usage. The latter can potentially be further evaluated by looking at the subjective results in section 5.3.1, where the participants provide feedback on both the size and location of the AR information. From the followed discussion in section 5.3.2, the AR effects in operation 2 and the coordinate system are especially objects in which the participants might not feel is justifying its space usage. The general UI could also be mentioned here, but it serves a broader purpose, which already has been discussed, and should thus be treated with more caution. If this is the case, the participants might not have been convinced that the way they view the world through the system is the best way to see it to solve the specific tasks at hand. They might be convinced in later evaluations if further work looks at the possibility to follow up some of the suggested component improvements from section 5.1.3 and 5.3.2. The reliability of some of the AR components should also be looked at to eliminate the need for on/off buttons in the general UI.

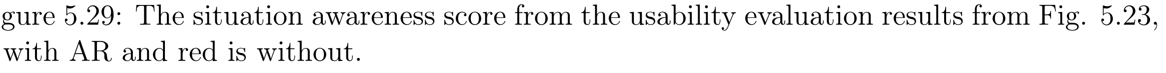


Figure 5.29: The situation awareness score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

Learnability
This experiment offers almost an acid test when it comes to learnability due to to the fact that the participants are busy students disposing their own studying time to test the developed system. A key factor for them when deciding to participate was that the tasks and system were easy to learn and fast to understand, so their time was not wasted on tedious introductions and tutorials. Moving over to the actual results, the learnability goal reaches a usability score of 4.5 for the AR system in Fig. 5.19 or 5.30, which makes it seems like they were satisfied with the process of getting familiar with the AR components and the AR system. This is also supported by the fact that they were given 10 minutes to read the introduction handout (appendix A) and get familiar with the system. Every participant said they were ready to start the operations before the 10 minutes had passed. Thus was the system, following what Nelson introduced in 1980, an easy to learn system, as it passed in every one of the eight occasions "The Ten-Minute Rule" which is a rule of thumb evaluating whether or not a system is easy to learn [42]. It is a very general rule, implying that a system is easy to learn if the users manage to learn how to use it within 10 minutes. Because this is a very general rule, it does not take the size of the system into account. A reason why everyone managed to understand it within 10 minutes, or at least thought they had understood it (with the one instance during operation 1 in mind, section 5.2.1), might be because of the size of the system is not very substantial. At the same time, however, this rule of thumb confirms that the system is not overly complex and nonintuitive which is important information to keep in mind during further development. It is also worth mentioning that the participants managed to get familiar with the AR system rather fast despite the fact that many of them had little prior experience or knowledge of AR, see Fig. 4.3.

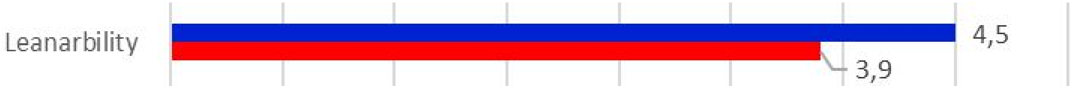


Figure 5.30: The learnability score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

Like situation awareness, learnability also scores higher in the general AR system evaluation compared to the AR relative to the non-AR case. It is actually a 0.6 gap between the two, see Fig. 5.30, which actually is quite significant. Why this might be, could potentially be due to the fact that the non-AR system is very simple. You have a screen with a camera feed presented on it. There is nothing more to it than that. So, by concept it is should be very easy to learn and understand. But then there is the question of, does it really provide what you want? Sure, it is easy to use, but does it enable you to use it the way you want in order to complete the operations? Probably not, however, this usability goal has its focus on the ease of use, and thus, when compared to the non-AR system it is understandable that some might rate the AR system more difficult to learn and get going with relative to the non-AR. This is really not so interesting though, as once you get familiar with the AR system, the range of functionality should outperform the non-AR one. Then it becomes a trade-off between a system that is very easy to use, due to limited functionality, versus a fairly decent system to learn and understand which provides a broader range of functionality which helps reduce the operation time as seen in section 5.2.1. The time invested to learn the system which provides more functionality should be earned back through reduction in operation time after some usages. Long-term (in a general setting with these results), the opportunity cost of not using the AR system would potentially be an increased amount of billable operator hours due to collectively longer operation times, and thus more expenses over time.

Seeing as the AR information seems to be rather easy to understand and get familiar with, another potential and much-used usability goal is memorability. This goal looks at how easy a system is to remember how to use [42]. This is kind off what we wanted to achieve with the general UI, bringing a familiar layout to make the system familiar and thus hopefully increase the memorability. Because of limitations both time wise and when it comes to bringing the participants back for another test, memorability is, therefore, not included in this report, but it rises as an interesting aspect to evaluate in further development.

Decision support

Last out, we look closer at the decision support usability goal, the score of 3.7 in Fig. 5.19 or 5.31, is rather low when this report sets out to introduce decision support capabilities to an operator interface. Fig. 5.20 gives somewhat of an explanation to this score. The participants are only rating the support during operation 1 to 2.5. Based on what's been discussed earlier in section 5.3.2, this does not come as a big surprise, but it is still a bit interesting that the highlighted maze profile did not result in a higher rating amongst the participants. The fact that the participants already had "solved" the maze when completing operation 1 with the non-AR system, might have reduced the value of having the maze presented in the next run. In addition, was the execution of the AR information in these operations, as discussed, probably not perfect, which also is indicated by looking at the results in Fig. 5.22. Here, only half of the participants felt the AR system supported them to faster understand how to execute the operations. So, even though earlier discussions in previous sections point out good decision support capabilities within the AR system when it comes to completing subtasks within the operations. Does these usability results indicate that the system has room for improvements when it comes to the overall support throughout the whole operation sequence. For the case of operation 1, this has been stated earlier and proved by the misunderstanding from participant 7, and should be used as a prof that a developers thought that the provided information should be sufficient doesn't always turn out to be correct when fresh users test the system. Therefore, this result can be viewed as another important reason for doing evaluations during the development of user experiences, and it serves as important feedback when trying to understand how to in a usable way introduce decision support in an AR operator interface.

Figure 5.31: The decision support score from the usability evaluation results from Fig. 5.23, blue is with AR and red is without.

Looking at the two other operations, the results are more promising. Different than for the subjective evaluation (section 5.3.1), operation 2 outscores operation 1. Based on the raised question about the necessity of the indicator in operation 2 from section 5.3.2, it seems like it is the confirmation regarding the cleanness of the marker which the participant's rates highly when it comes to support in operation 2. This is the kind of supportive information which cuts the time spent on being unsure about something and then have to go back and check in order to make a decision, which in this operation meant going back to the marker to check if it indeed was clean. This implies decision support capabilities within a subtask in operation 2.
Moving on to the last operation, there is no doubt amongst the participants. They have rated the support to 5 out of 5. It should, therefore, be reasonable to draw the conclusion that for operation 3, the system did indeed support the participants when they had to come to a decision of what to do with the two different valves. Of course in this setting, it was possible to construct the operation in a way that an AR system, containing all relevant information, really could shine. From what the participants told during the experiment, some of them felt like they would be told by the system in operation 3 if they did something wrong or not clever. This goes in the direction of producing the feeling you want to achieve with a system that does decision support in a great way. You want your users to be comfortable, confident and effective when using the system, something decision support can come a long way in providing. Essentially you want to reduce the time spent on being unsure of what to do during operations. For this, you need a highly reliable system that understands the environment and learns from previous situations. To achieve this learning aspect, further work could be looking into how artificial intelligence could be used to create a system that learns from previous situations and actions. At the same time should operation 3 be used as the standard in order to achieve an all over great decision support system, not a system that is decent in some situations and excellent in others.

Another aspect of receiving decision support from the system is what the AR depth tools bring to the table. Having information about the depth field ready at the disposal should help the participants in deciding when they are at the right position to stab and when not to. From Fig. 5.13 and 5.14 one can see that the system supports the participants with that. Thus the decision support aspect is helping them to increase their effectiveness during operations. The fact that effectiveness scored so high is probably part of the reason why the decision support usability goal scores so high when evaluating the AR system relative to the non-AR, see fig. 5.31. Another might be a feeling of being alone when executing without AR, and that you are accompanied by someone or something while using AR. This someone or something could feel like the "person" behind the system which helps you along, acting as a kind of instrumental support to the participant [128]. If this is the case, the AR system could also be bringing more well-being to the participants. Something that is important in a real corporate setting.

From the results presented, the potential of supporting subsea ROV operators when decisions are to be made should be considered beneficial. To have such a powerful tool at the disposal in an industry where operation time is related to high costs, could definitely result in accumulated savings over time. In addition, could mistakes or uninformed decisions in this industry, in worst cases, lead to catastrophically consequences both for our nature and financially for the broad spectrum of stakeholders as described in section 2.2.1. This only emphasizes the potential of improvement within safety and cost-saving decision support capabilities within an AR operator interface could introduce.

Throughout this section, we have discussed different reasons to why the usability scores turned out as they did. Overall it must be said that the AR operator interface scores high evenly throughout every usability goal. However, some shortcomings have been discussed and recommendations on how to approach these in further work have been proposed. There is also some aspects of the system which scores very high in the evaluation. It is important that these are not forgotten during further development as changes to other areas without them in mind, could substantially reduce their scores in later evaluations. The experience earned from the implementation methods used to achieve these high scores should rater be used as guidance to increase the overall usability score. With that being said, even though there were some shortcomings, there were no situations or indications discovered that indicated that the implemented decision support and depth perception capabilities of the AR operator interface were not usable.

5.3.5 Results from the system usability scale

The survey used in the experiment evaluation (section 4.6.1) also contains the 10 questions used in the system usability scale questionnaire, introduced in section 2.3.3 and shown in Fig. 2.12. These questions have five response options ranging from strongly agree to strongly disagree. In Fig. 5.32 the SUS results are presented and strongly agree is marked with the darkest blue, while the other end of the response scale is marked with the darkest red. This figure also includes the 10 questions used in the SUS evaluation and shows what percentage of the participants answered the response with the corresponding color. From this figure, one can see how the participants rated the system based on the individual questions, and in most cases, the user answered the two answers most in favor of the system. There are however some exceptions. The second and sixth question (from the top), which looks at the confidence while using the AR system and how well the AR is integrated, is the only questions with responses at both ends of the scale (i.e., disagree and agree). At the same time does the fifth question, which is about inconsistencies in the system, also stand out because of the high number of neutral responses. The remaining questions are mainly characterized by positive sided responses, but also with a few events of neutral responses.

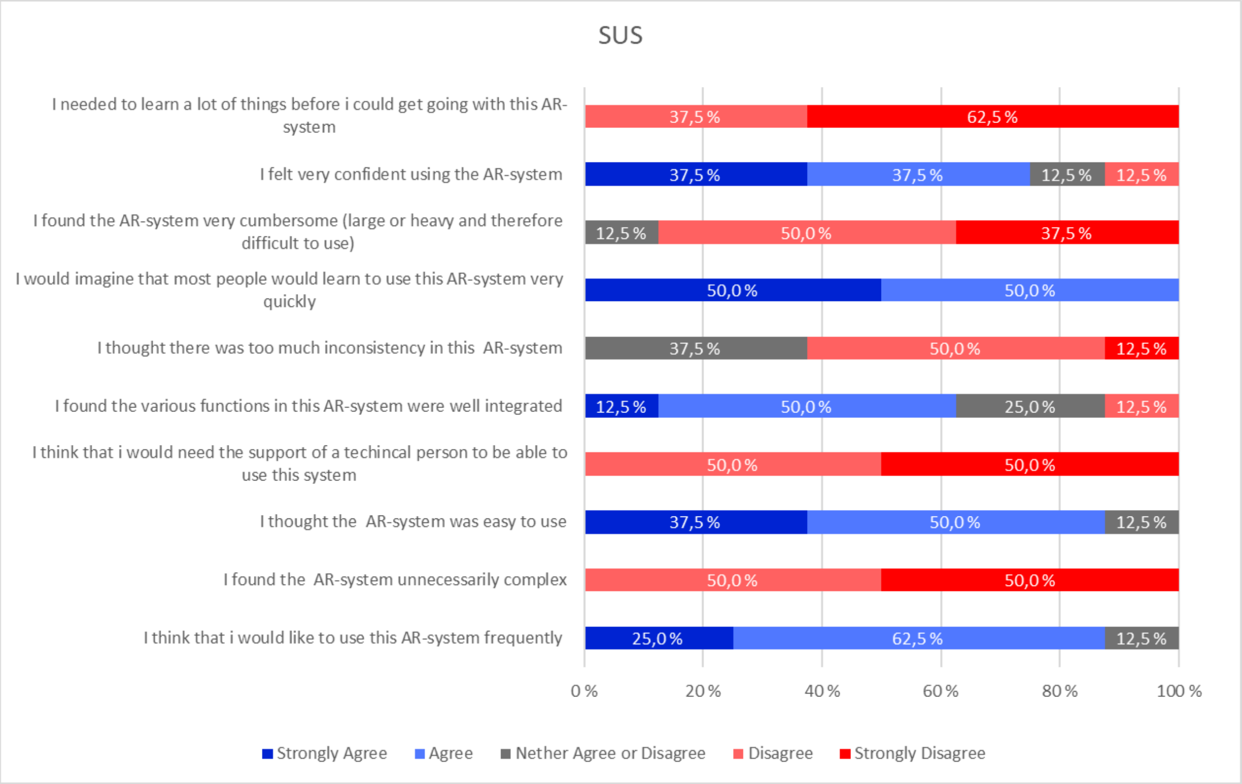


Figure 5.32: The evaluation results of each of the AR components versus each other.

To better understand the results from the SUS questionnaire, the system usability scores for each user is calculated. This is done through the steps given in section 2.3.3. The resulting scores after the calculations are listed in Table 5.6. Following the SUS score scales presented in section 2.3.3, the average SUS score is just above the limit to score a B which means the average score is far within the general average SUS score of 68. When comparing this result with the SUS scores in trymyUI's [129] database, a score of 80.3 manages to just place itself amongst the top 25% percentile. In other words, this system rates better than 3 out of 4 systems tested in their database. In Fig. 5.33 the average SUS score is marked in the SUS scoreboard presented in section 2.3.3. Other than scoring a B, this figure shows how the AR system is rated as good and thus within the acceptable criteria on Bangor, Kirum, and Miller's proposed scales [62]. Looking at the participants individually in Table 5.6, five of the participants has high enough scores to get a B, meanwhile one participant manages an A followed by the two least scoring with a C and a D.

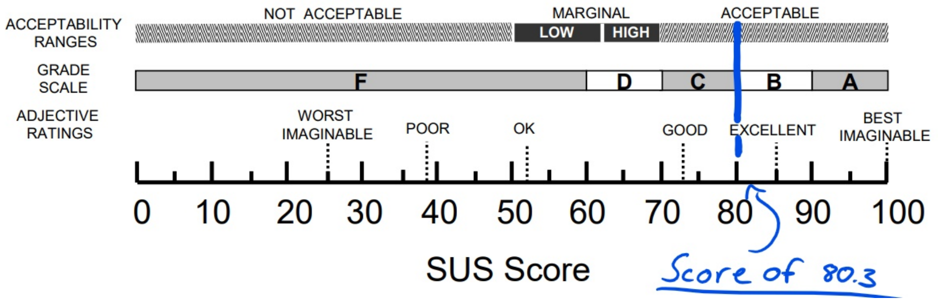


Figure 5.33: The results from this SUS highlighted in Bangor, Kortum, and Miller's [63] SUS scoreboard.

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Score**
1	4	2	3	2	3	3	4	1	2	2	**65**
2	3	1	5	2	4	2	5	2	5	2	**83**
3	4	1	4	1	3	3	5	2	4	1	**80**
4	4	2	4	1	4	2	4	1	4	2	**80**
5	5	1	5	2	5	2	4	2	5	1	**90**
6	4	1	4	2	4	1	5	1	4	1	**88**
7	4	2	4	1	4	3	4	2	3	1	**75**
8	5	2	5	1	2	2	5	3	5	1	**83**
Average	**4.1**	**1.5**	**4.3**	**1.5**	**3.6**	**2.3**	**4.5**	**1.8**	**4.0**	**1.4**	**80.3**

Table 5.6: Table of the SUS responses from each participants and their calculated SUS score.

The results following the proposed calculations for learnability and usability scales by Lewis and Sauron in[60], is presented as a graph in Fig. 5.34 and in Table 5.7. These scores are calculated through the guidelines introduced in section 2.3.3. In the graph, every participant's SUS score is presented versus the corresponding calculated usability and learnability scores. From this almost every participant seems to score rather evenly throughout the three different classifications, however, three of the participants stands out with a top score of 100 in learnability. These three seems to score lower on the two other classifications compared to the rest, but this is compensated for with the excellent score in learnability.

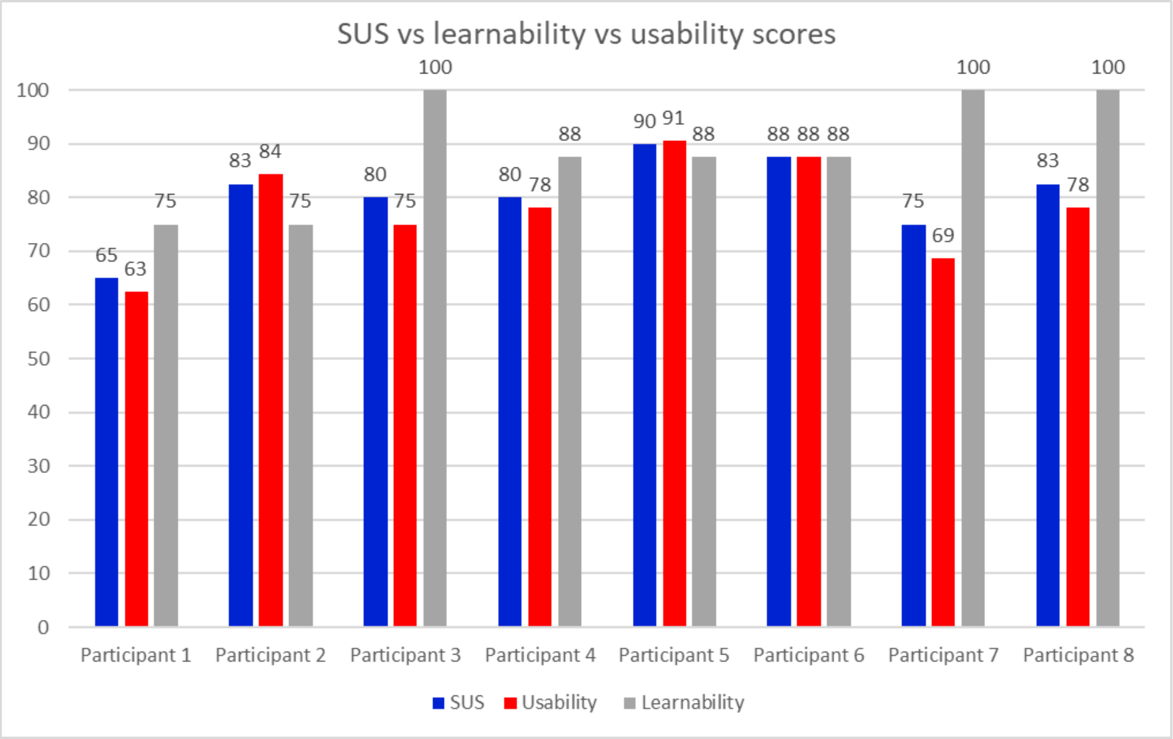


Figure 5.34: The evaluation results of each of the AR components versus each other.

Type of score	Avg	Var	SD
SUS	80.0	52.25	7.23
Usability	78.1	78.12	8.84
Learnability	89.1	95.21	9.76

Table 5.7: The average (avg), variance (var) and standard deviation (SD) of the three different score calculations.

5.3.6 Discussion of the system usability scale results

The system usability scale evaluation comes as an addition to the usability evaluation presented and discussed earlier in this chapter. In this section, the SUS results will be discussed to see if any usability problems are discovered, and to relate the usability of the AR operator interface to other SUS evaluated systems.

To better illustrate how every participant responded to each of the 10 questions Fig. 5.32 was created. It is here, as mentioned above, fairly straight forward to see that questions 2 and 6 (from above) stands out with both red and blue responses (and gray). This indicates some disagreement among the participants about the felt confidence and the integration of AR. One of the objectives of adding decision support capabilities is to increase the participant's confidence during the operations. Seeing as confidence stands out here and decision support did the same in the usability evaluation, there could be aspects of the AR system that prevents it from utilizing its full decision support potential. By looking at the results from the SUS evaluation one could think that the confidence and thus the decision support is potentially affected by the level of inconsistencies. Because, when using a system with lacking consistency, which might come from AR objects that are not perfectly integrated, the users trust in the system might be reduced and thus the level of confidence as well. This thought is reinforced when looking at the responses to question 5 (from above) which asks about inconsistencies in the system. Almost 40% of the participants neither agree or disagree with the thought that there was too much inconsistency in the AR system. It is, therefore, reasonable to think that the difficulty of integrating AR objects in the real world could have an influence on the user's confidence and thus the felt decision support during operations. This challenge of implementing AR has already been introduced in section 2.1 and discussed in section 5.1.3, giving the indications that this is one of the important areas to focus on improving when going forward with the desire to design and produce a robust, reliable and decision supporting AR operator interface.

When looking at the results from the completed SUS, it is interesting to see how high the average SUS score rates. Taking a further look at the scoreboard presented in Fig. 5.33 it seems like the participants thought the tested AR system was good and an acceptable tool when solving the operations. At the same time, it is important to emphasize how this system was tested under controlled circumstances with carefully selected operations to highlight the potential of the system. Because of this ideal setting, it would have not been totally crazy to achieve a score close to 100 when testing the system in operations it was designed to perform in. This again proves the use for usability evaluations, as one can detect limitations and usability problems of the developed system. One of these is mentioned above, and several more are discussed above in the rest of this chapter.

By looking at the Table of SUS responses in Table 5.6, one can see that some participants are more critical to the system than others. However, most scores are between 80 and 90, but with a few outliers, which shows the importance of having a solid number of participants to have enough responses to get a representative average. But the reason for participant 1's low score of 65, is unsure, the person might not be conveniences by the AR-system? Or might have expected more from it? What this score does though, is reminding the developer of the difficulty of creating a system, application, product, etc., that everyone finds very usable. While we are at the subject of the participants, an interesting thought in the context of creating an AR operator interface aimed towards subsea ROV operators. Is that it would be very interesting to have these subsea ROV operators go through a test experiment with the AR system to see how they score on the scoreboard. Because, in the big picture, these are the desired users. So, for further work, the idea of having subsea ROV operators as participants is very intriguing and their usability feedback would be very relevant, even though the student's feedback in this report helps immensely in the current development phase of the system.

From the calculations proposed by [60] the results in Fig. 5.34 and Table 5.7 emerges. It is here interesting to see how the SUS score represents both usability and learnability. From the figure and table, one can see that it is actually the learnability aspect which is the source to the highest scoring areas in the SUS questioner, and usability is the weaker part. This is not so surprising when looking at the result provided in the usability evaluation in section 5.3.3. Here, learnability is one of the highest scoring usability goals, while the general usability is dragged down by lower scoring usability goals. So, from this, it seems like the SUS scores reflect the scores from the usability evaluation, which is great when keeping in mind that one of the goals with the SUS questioner was to put the usability of the developed AR operation interface in context with other user experience dependent systems. Other than this, there is not much else to extract from these results, other than previously discussed topics in this chapter. The results do not indicate any major usability problem, however, it suggests that the participants rated the usability of the AR system as good and that the learnability aspect of the system is probably one of its stronger characteristics.

Chapter 6

Conclusion and further work

The report is finalized with a conclusion followed by suggestions for further work and improvements. The conclusion seeks out to answer the research question presented in section 1.2, while the further work section presents different recommendations, areas to improve, and finally suggests an area to focus new research questions within.

6.1 Conclusion

In this report, the research question is investigated through theoretical, practical, and experimental methods. An AR operator interface has been implemented with AR depth tools and other AR components to provide depth perception and decision support capabilities to the system. Even though there have been some roadblocks on the way and difficulties with integrating the AR layer onto the physical world, a group of participants has used the implemented system in an experiment to evaluate this contributed AR operator interface. Based on the implementation results and the experimental results, we can conclude that emerging technology within AR can be used to introduce depth perception and decision support capabilities to a 2D monitor based interface between an operator and an operated object. However, we have found that the quality and performance in which some of the AR components manage to utilize these capabilities vary. Some manage to deliver this information in good and great ways, while others only do a decent job. From these findings, we can also conclude that the depth bar is the depth tool which does the best job in providing depth perception to the users of the system. While an interactive user experience appeared to do the best job of providing decision support.

In addition to just deliver depth perception and decision support, it was important, in this report, that this was done in a usable way. So, from the focus on usability throughout large portions of the evaluation within the experiment and subsequent results, the participants have rated the usability of the developed AR system as good. Compared to a large database of evaluated systems, does our AR system score better than $\frac{3}{4}$ of the evaluated systems.

As the AR operator interface is designed and developed with depth perception and decision support in mind, we can conclude based on the good usability ratings, that these capabilities are indeed integrated in a usable way into the AR system.

There are too many limitations within this report to conclude that this AR operator interface can be used as the interface between subsea ROVs and their operators. However, throughout the analysis, some of these limitations have been discussed and some recommendations on how to potentially free the system from these limitations have been presented. Further, the quantitative experiment results should not be overlooked, as these indicate that the AR system increases both effectiveness and efficiency during operations. The potential cost savings from reduction in operation time and general increased well-being of the operators should be used as motivation for further research and development to, hopefully, one day overcome all of these limitations and end up with a system deployable within the interface between subsea ROVs and their operators.

To finalize the conclusion of this report, we have contributed with a system that does indeed manage to use AR technology to provide increased decision support and depth perception information, to an operator, operating an object, by looking at a video feed on a 2D monitor. The way in which the system provides this information is through good usability, based on the experimental results. Even though this system is not ready to be deployed as the interface towards a subsea ROV, it documents promising findings and showcases an alternative approach to improving work class ROV operations, with more focus on the human aspect, which previously has not seen so much research focus. So, hopefully, this paper motivates and functions as an eye-opener to increase the interest around the research question of this report.

6.2 Further work and suggested improvements

As not much previous research is found on developing an AR operator interface intended for subsea use, we pretty much started from scratch in this report. With this comes a set of proposed recommendations for further work, to continue the work with the research question within this report. In addition, suggested improvements are also provided, as throughout the experiment you kind of get a new perspective on the development as you put it in the hands of other users (i.e., the experiment participants).

One immediate recommendation to do when continuing the work is to try to investigate more into the area of which this report does not manage to conclude. That includes trying to get hold of some subsea ROV operators who can test the AR system. Hopefully, they can provide highly relevant input on improvements, suggestions, and features they would like to have in such an interface. They are after all the intended end users. Also, methods to free the system from some of its limitations should be investigated in future work. When looking at the markers this could include moving over to markerless tracking. This could potentially be very demanding to get properly functioning in harsh subsea environments. Utilization of research done to do marker tracking underwater then arises as a potential alternative for

an improved marker-based solution. This would essentially introduce a tracking system one have more ownership of and thus freedom to integrate desired features to better integrate AR information. This is something seen from the results presented in this report, that one of the developed AR systems shortcomings is the way some of the AR objects appear in the operator interface. In further work more emphasize should thus be directed towards improved alignment of AR components onto the physical world. These components size compared to the amount of information they provide should also be considered, in an attempt to improve size and location of some AR components, as this also stands out as one of the AR systems weaknesses. This could be improved by using more hours on perfecting the Vuforia and Unity execution, or potentially through other system setups with an own developed tracking system, or by seeking help from additional sensors to improve pose estimation and tracking through a sensor fusion system.
If work is directed towards tracking and AR integration, as described above, the robustness and reliability of the AR information are expected to improve. As found in this report, the degree of decision support, as well as the usability, is then also expected to improve. This supports the case for focusing further work within this area.

When it comes to the depth tools, improvements should be done to the depth bar in order to make it more functional when the operated tool moves past the target, we have already suggested a potential way to do this within the report. The general potential of the path is also very interesting. If further work and inspiration from adaptive path solutions could enable the path to recommend the best route to a target, while taking obstacles into account, it would provide much value to the operators and operations because of collision avoidance and potentially other important factors. It would also be a great leap towards fully autonomous navigation of the work class ROV.

A general recommendation for further work is the recommendation to have smaller usability tests frequently, to regularly collect feedback from users on their opinion on the usability of the system. Memorability is then an interesting usability goal that should be a considered addition to the list of usability goals. It is also important to remember that every change to the appearance, contents, and location of information within the operator interface could affect every usability goal in both positive and negative directions. This is certainly the case when adding new features to the system, which is essentially why frequent usability evaluations are recommended.

As the research question in this report has been investigated, a suggestion for a new question to investigate in further work would be to include superimposition of maintenance instructions and assisting information into the AR operator interface (see [125] for inspiration). This is also an area where AR can provide great support to the operators and it opens the door for having experience sharing amongst operators and engineers without the need for their physical appearance.

Appendix A

Participant handout

Augmented reality vs no augmented reality user experiment – Participant information hand-out

Content of experiment

1) Read instructions and ask questions if something is unclear.
2) Execute the three operations with the non-AR system
3) Execute the same three operations with the AR system
4) Finish the experiment with a subjective evaluation

The test rig

The rig consists of a wooden stick, called the tool, which the operator (the participant) can control with 5 degrees of freedom. This is due to the ball joint in the tool mount. The tool can be dragged in and out of the mount by dragging or pulling the stick while holding the pin on the top of the tool mount. Further, the stick has two types of tools. In operation 1 and 2, there will be a horizontal extension at the end of the stick giving the stick a L profile. While in operation 3 the stick will only consist of the vertical part, giving it a I profile.

There are big walls blocking the operator's sight of view. This is because the tool is supposed to be operated and maneuvered by looking at the video feed presented on the conveniently placed computer.

Behind the walls there are three different scenes. Each scene represents one operation.

The program running on the computer consist of a main menu with 4 choices. First is the whole program from the non-AR part, while the remaining three are for each of the three AR operations. So when one of the AR operations are executed, the participant should hit main menu before starting the next operation.

Operation 1 – Complex stabbing operation

In operation 1 the goal is to insert the tool in the middle of the orange target. The orange target has unknown profile and consists of walls which makes it more difficult to reach the middle of the target. Enter the target on the lower section all the way back of the orange part and navigate yourself to the middle to complete the operation.

Operation 2 – Cleaning marker/tool by "removing" dirt

In the scene the operator will see cubed profiled hook. The marker can be cleaned by inserting the tool into the cubed hook and then poll down. To ensure that the marker is completely clean, the operator should move close to the marker and inspect and confirm that it is clean. When the operator tells the evaluator that the marker is clean, the operation is completed.

Operation 3 – Valve readings

During the non-AR run, the operator should maneuver the tool so that it is possible to read the valve position of valve 1 and 2. Thereafter these levels should be compared with the planned positions in the operation list. After comparing the positions, an action for both valve 1 and 2 should be vocally delivered to the evaluator.

For the AR part, the system offers interaction-based information. The tool should be docked, and the operator should be able to control the valves through the AR user interface on the computer.

Operation list 10.04.2019

Valve 1 (well choke)		**Valve 2 (well choke)**		**Manifold choke**	
Time	Valve position	Time	Valve position	Time	Valve position
00:00	60%	00:00	50%	00:00	90%
01:00	60%	01:00	50%	01:00	90%
02:00	50%	02:00	50%	02:00	90%
03:00	50%	03:00	30%	03:00	90%
04:00	50%	04:00	30%	04:00	90%
05:00	40%	05:00	30%	05:00	80%
06:00	40%	06:00	70%	06:00	80%
07:00	40%	07:00	50%	07:00	80%
08:00	40%	08:00	50%	08:00	50%
09:00	30%	09:00	40%	09:00	50%
10:00	30%	10:00	30%	10:00	50%
11:00	50%	11:00	30%	11:00	50%
12:00	50%	12:00	50%	12:00	70%
13:00	60%	13:00	40%	13:00	70%
14:00	60%	14:00	40%	14:00	70%
15:00	50%	15:00	60%	15:00	70%
16:00	70%	16:00	50%	16:00	80%
17:00	60%	17:00	40%	17:00	80%
18:00	60%	18:00	30%	18:00	90%
19:00	60%	19:00	30%	19:00	90%
20:00	60%	20:00	40%	20:00	90%
21:00	40%	21:00	50%	21:00	90%
22:00	30%	22:00	50%	22:00	90%
23:00	30%	23:00	40%	23:00	90%

Operation database for well-chokes on Gullfaks B

Valve number	Manifold tied to	Cluster	Last update	Additional operation information	Information last updated by
5	1	SW	03.04.19	Normal operation	Ola Nordmann
6	1	SW	03.04.19	Normal operation	Ola Nordmann
7	1	SW	25.03.19	Out of operation due to intervention. Expected back in normal operation 12.04.2019	Kari Nordmann
1	2	NW	10.04.19	Valve position Changed to 20% due to high upstream pressure, set valve position according to operation list after 09:00	Kari Nordmann
3	2	NW	08.04.19	Normal operation	Ola Nordmann
2	3	E	10.04.19	Valve position overridden to 70% at 8 o'clock this morning by chief of operations.	Kari Nordmann
4	3	E	02.02.19	Operates in fully open position due to low pressure in the well	Ola Nordmann

Valve operation guidebook

1) If manual reading of valve position is the same as the position in the operation list. No further action needed.
2) If manual reading of valve position is not the same as the position in the operation list
    a. Check operation database for additional information
    b. Contact the person who logged the information if needed.
    c. When the needed information is collected, make a decision on what to do about the valve position
3) If chief of operations overrides a valve, contact this chief to clarify the situation and then decide what to do.

Decide what to do with valve 1 and 2 to complete the operation

(Check one of the boxes for each valve and fill in the blank if any)

Valve 1

☐ No action needed

☐ Change valve position to __________ %

☐ Contact ____________________________ to seek more information

Valve 2

☐ No action needed

☐ Change valve position to __________ %

☐ Contact ____________________________ to seek more information

Appendix B

3D CADed and printed 3D models

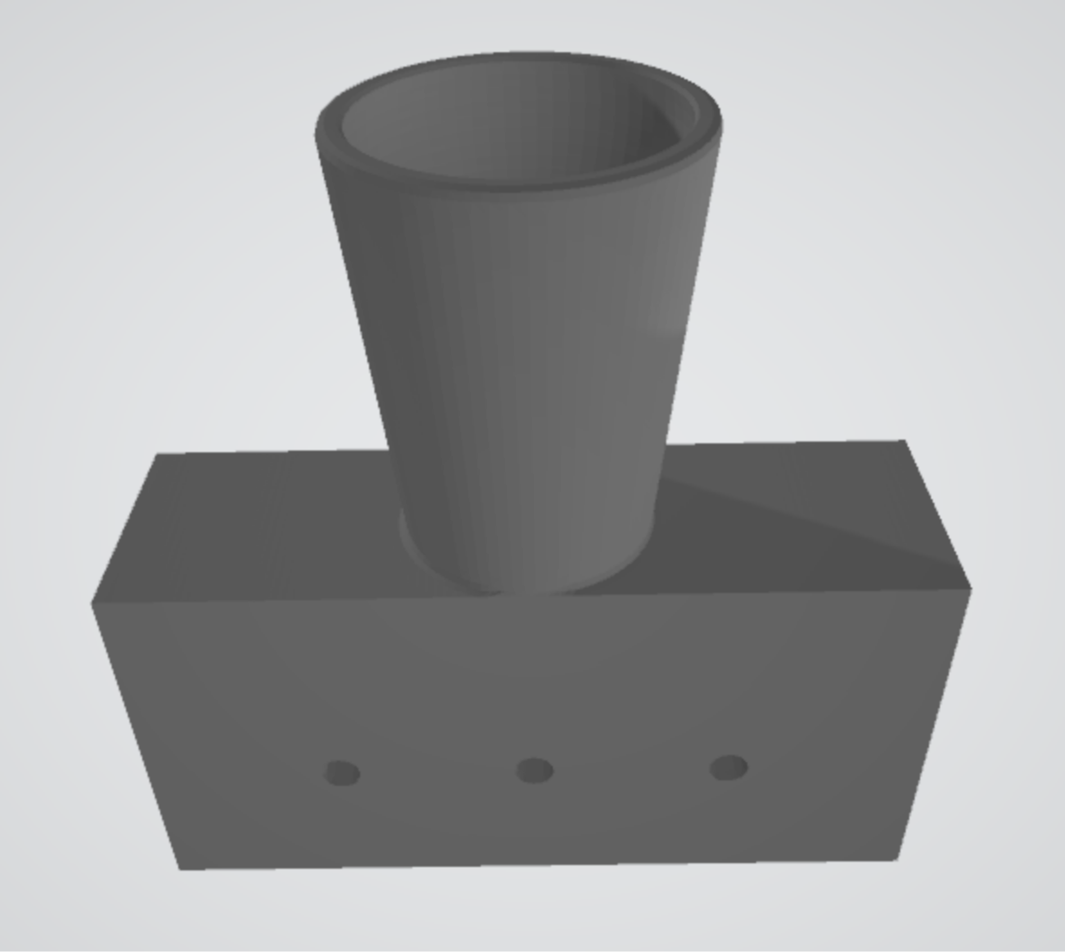


Figure B.1: A 3D model of the 3D printed female part of the ball joint.

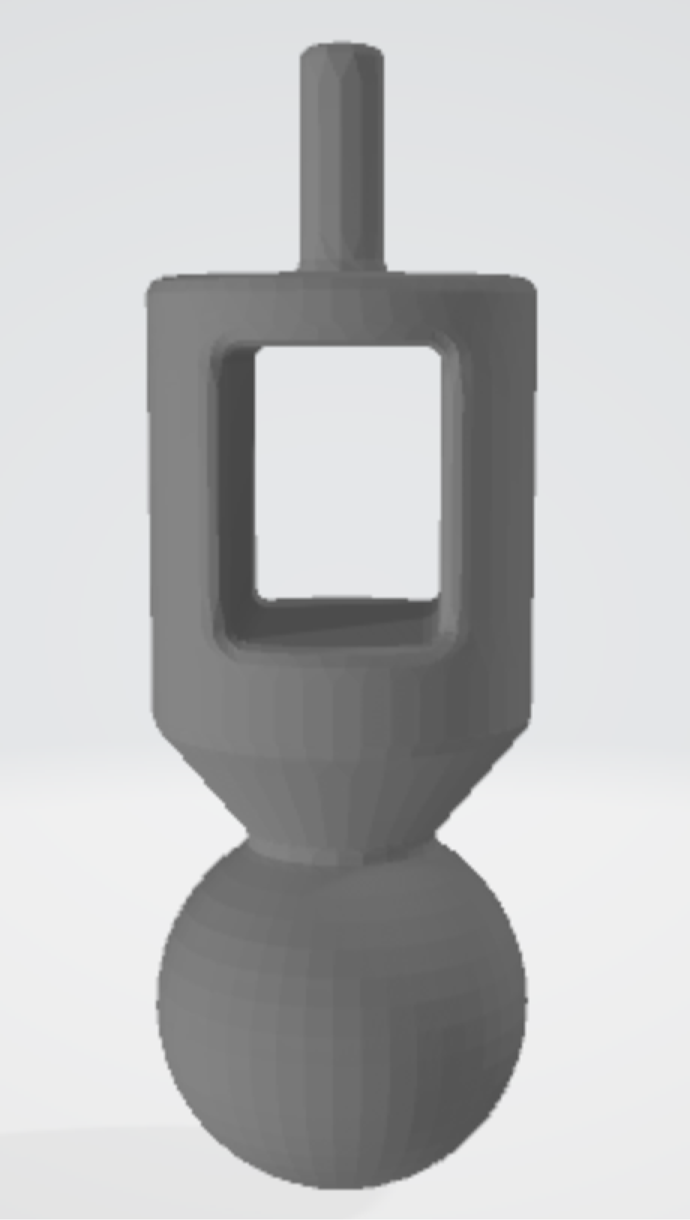


Figure B.2: A 3D model of the 3D printed male part of the ball joint.

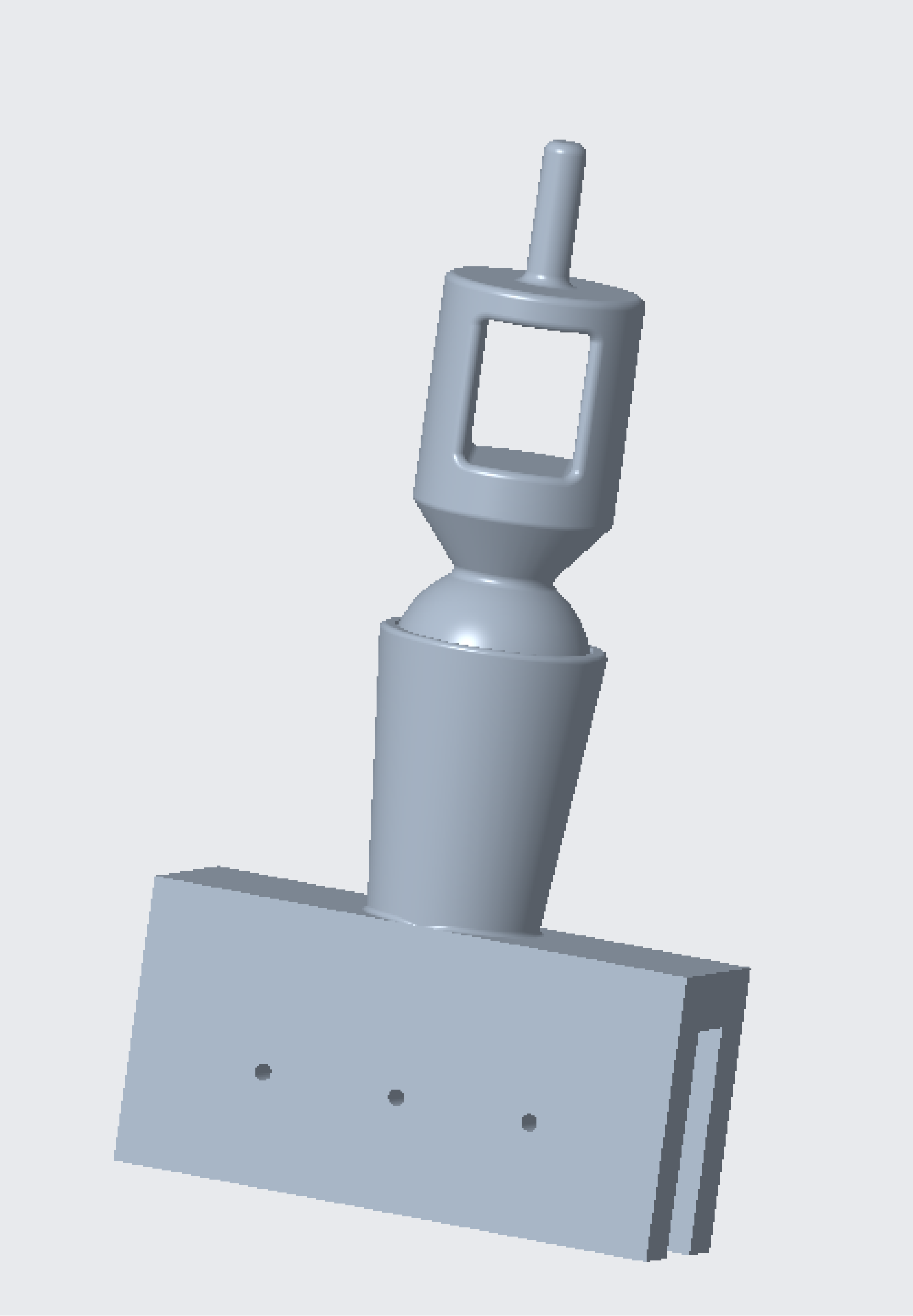


Figure B.3: A 3D model of the assembly of the contributed ball joint.

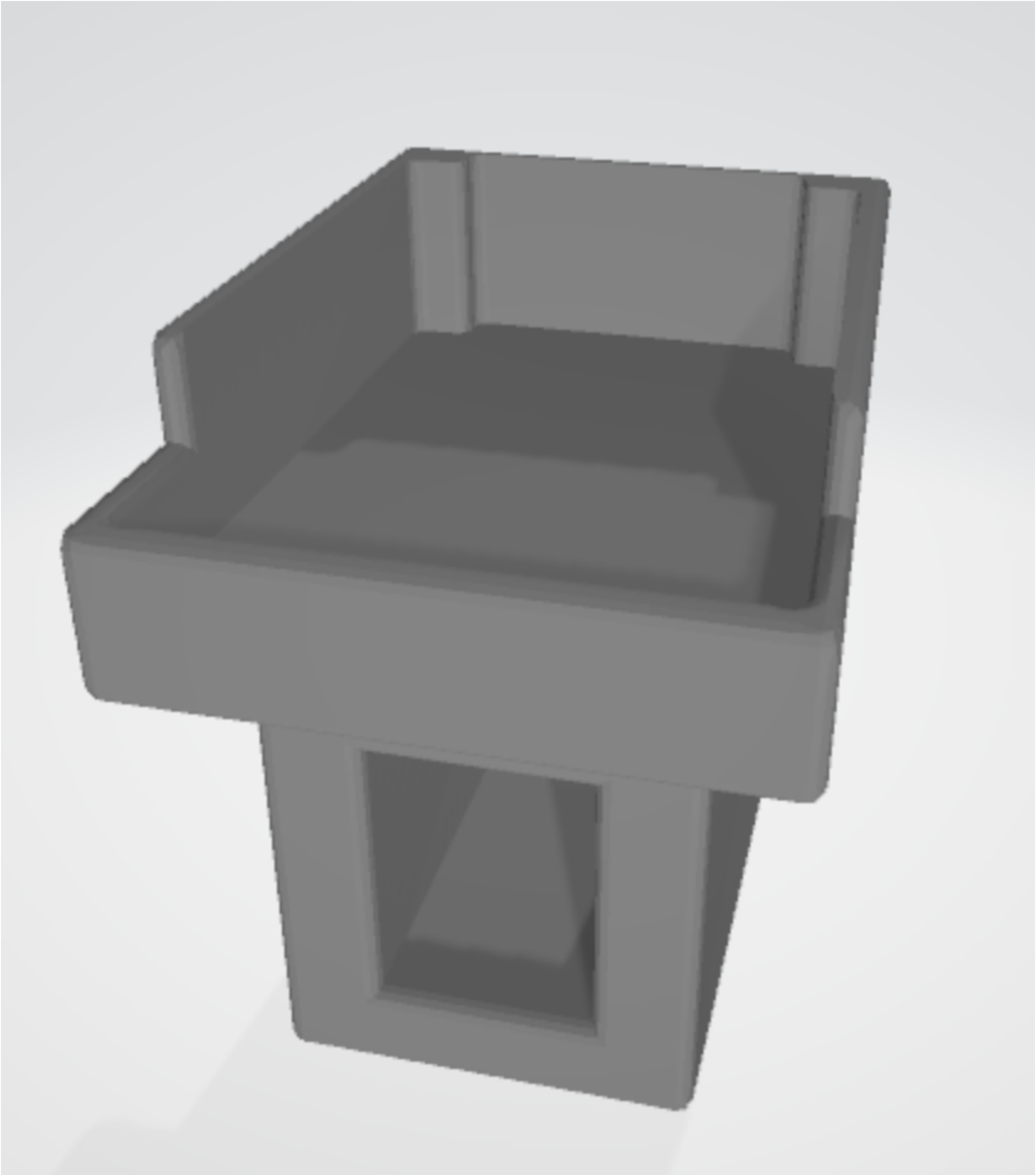


Figure B.4: A 3D model of the camera mount which sits right on the vertical beam of the tool, used during operation 3.

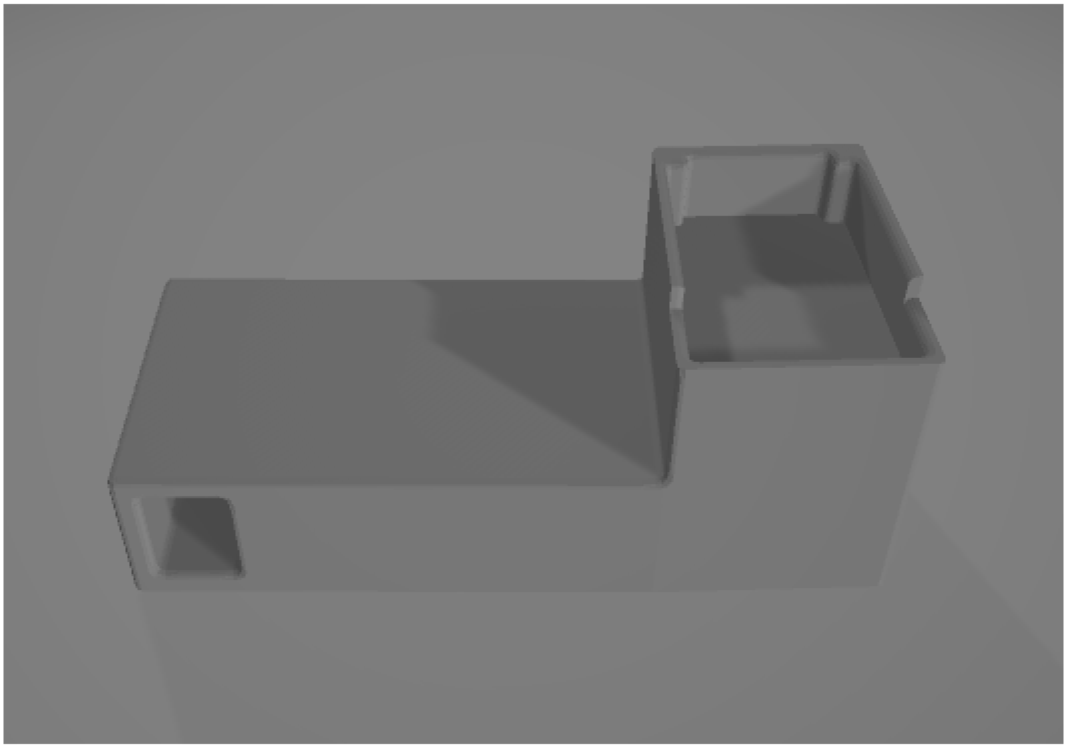


Figure B.5: A 3D model of the extended camera mount which moves the camera position to the left.l, Used during operation 1 and 2.

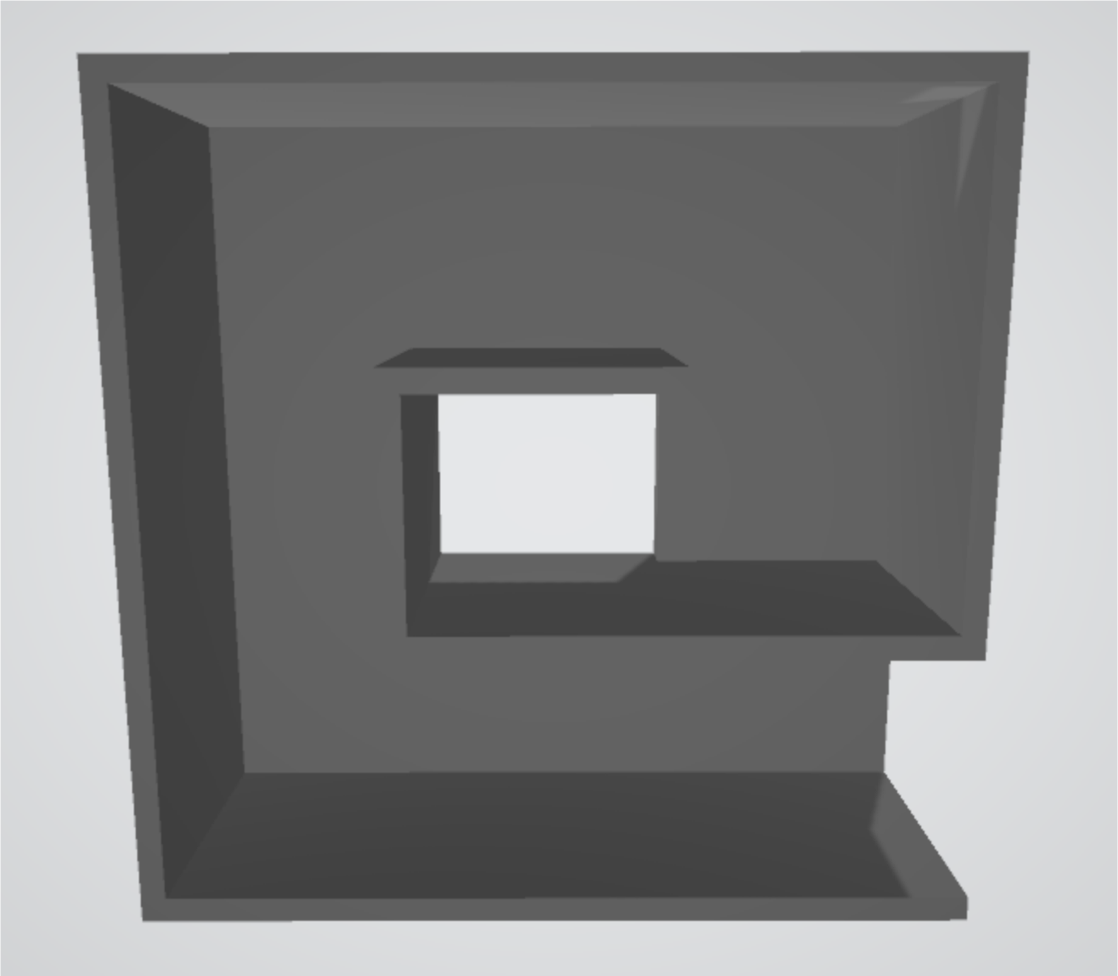


Figure B.6: A 3D model of the 3D printed maze used as a component within operation 1.

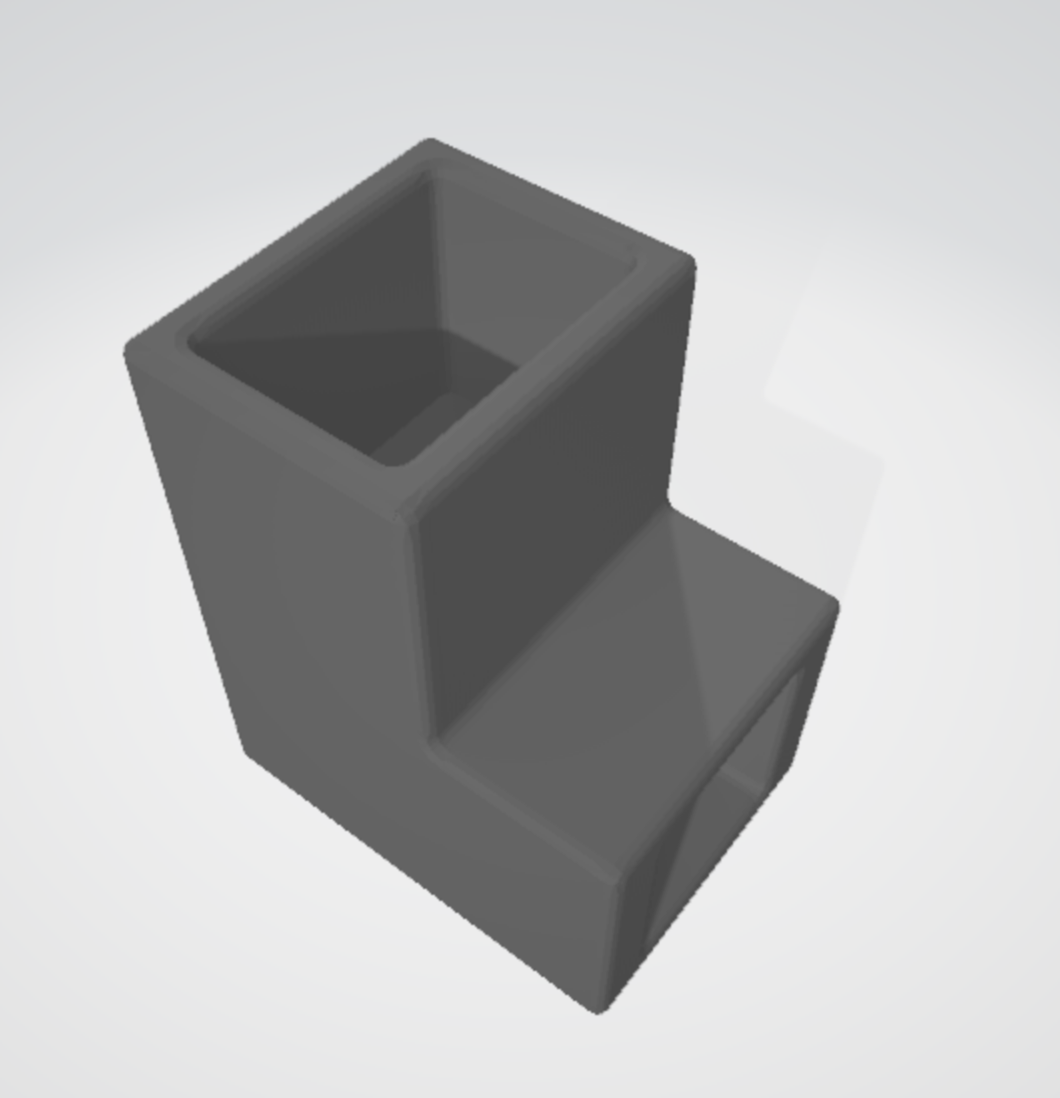


Figure B.7: A 3D model of the 3D printed toll extender part. Used to attach and detach the horizontal beam extensions to the tool.

Figure B.8: A 3D model of the 3D printed docking used to enable the tool to be docked in between operations and during operation 3.

Appendix C

C# scripts used in Unity with Vuforia

Depth bar

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class DepthBar : MonoBehaviour
{
    [SerializeField]
    private Transform marker;
    [SerializeField]
    private Transform ARCam;
    public float dybde;
    private float ToolPoseY;
    private float TargetPoseY;
    private float poseY;
    public Slider mSlider;

    // Update is called once per frame
    void Update()
    {
        ToolPoseY = ARCam.transform.position.y - 63;
        TargetPoseY = marker.transform.position.y;
        poseY = -(TargetPoseY - ToolPoseY);
        dybde = poseY;
        mSlider.value = dybde;
    }

}
```

Listing C.1: The depth bar C# script

Script for operation 1

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Hot_stab : MonoBehaviour
{
    // Pose variables
    [SerializeField]
    private Transform marker;
    [SerializeField]
    private Transform ARCam;
    private float poseX;
    private float poseY;
    private float poseZ;
    private float pose;
    private float ToolPoseX;
    private float ToolPoseY;
    private float ToolPoseZ;
    private float TargetPoseX;
    private float TargetPoseY;
    private float TargetPoseZ;
    private float WaterDepth;
    [SerializeField]
    private GameObject operation1;
    [SerializeField]
    private GameObject tool1;
    private float startTime;
    private int count;

    // Depth bar
    public Slider mSlider;

    // Path
    public LineRenderer path;
    public GameObject pathHolder;
    public GameObject barfix;
    private Vector3 ToolEnd;
    [SerializeField]
    private GameObject toolMarker;
    //public int X;
    //public int Y;
    //public int Z;
    private float MarkerPoseX;
    private float MarkerPoseY;
    private float MarkerPoseZ;
```

```
private float EstX;
private float EstY;
private float EstZ;
private int i;

// Path colors
public Color Green = Color.green;
public Color Red = Color.red;
public Color Blue = Color.blue;

//Path Filter
private List<float> PoseEstX = new List<float>() { 0, 0, 0, 0, 0 };
private List<float> PoseEstY = new List<float>() { 0, 0, 0, 0, 0 };
private List<float> PoseEstZ = new List<float>() { 0, 0, 0, 0, 0 };
private int NumerOfPoseEst;

// Text variables
[SerializeField]
private TextMesh poseX_text;
[SerializeField]
private TextMesh poseY_text;
[SerializeField]
private TextMesh poseZ_text;
[SerializeField]
private Text barY_text;
[SerializeField]
private TextMesh kord_X;
[SerializeField]
private TextMesh kord_Y;
[SerializeField]
private TextMesh kord_Z;
[SerializeField]
private Text WaterDepthText;

// Start is called before the first frame update
void Start()
{

    Vector3 ARcamPose = new Vector3(0, 0, 0);

    // Configure the path
    path.startWidth = 0.35f;
    path.endWidth = 0.35f;


    // Highlight operation
    operation1.SetActive(true);
```

```
        tool1.SetActive(true);

        count = 0;

    }

    // Update is called once per frame
    void Update()
    {

        ToolPoseX = toolMarker.transform.position.x - 13;
        ToolPoseY = toolMarker.transform.position.y;
        ToolPoseZ = toolMarker.transform.position.z;

        MarkerPoseX = marker.transform.position.x - 15;
        MarkerPoseY = marker.transform.position.y + 14;
        MarkerPoseZ = marker.transform.position.z - 7;

        // Calculate the relative pose
        poseX = -(MarkerPoseX - ToolPoseX);
        poseY = -(MarkerPoseY - ToolPoseY);
        poseZ = -(MarkerPoseZ - ToolPoseZ);


        Vector3 Pose = new Vector3(poseX, poseY, poseZ);


        // ************ Path and Pose filter

        // Add pose estimates to list
        PoseEstX.Add(poseX);
        PoseEstY.Add(poseY);
        PoseEstZ.Add(poseZ);

        i = PoseEstX.Count;

        //    // Get the average of the 4th last poses
        foreach (float value in PoseEstX)
        {
            EstX = (PoseEstX[i - 4] + PoseEstX[i - 3] + PoseEstX[i -
               2] + PoseEstX[i - 1]) / 4;
            EstY = (PoseEstY[i - 4] + PoseEstY[i - 3] + PoseEstY[i -
               2] + PoseEstY[i - 1]) / 4;
            EstZ = (PoseEstZ[i - 4] + PoseEstZ[i - 3] + PoseEstZ[i -
               2] + PoseEstZ[i - 1]) / 4;

        }
```

```
// Set the origin of the path at the tip of the tool
path.transform.position = new
   Vector3(toolMarker.transform.position.x - 18,
   toolMarker.transform.position.y,
   toolMarker.transform.position.z);

//// Color of path

// Color change based on distance in y-direction from target

// Convert distance between tool and target to a 0f and 1f
   range
float lerp = Mathf.InverseLerp(0, 60, poseY);
float newinterval = Mathf.Lerp(0f, 1f, lerp);

// Lerp the color between far and near
Color lerpColor = Color.Lerp(Blue, Red, newinterval);
path.GetComponent<Renderer>().material.color = lerpColor;


//Define the 3D line from the avrage of pose estimates for a
   more stable line representation.
Vector3 pathX = new Vector3(0, EstY, 0);
Vector3 pathY = new Vector3(0, EstY, EstZ);
Vector3 pathZ = new Vector3(EstX, EstY, EstZ);

// Assign the vector lines to the line renderer
path.SetPosition(1, -pathX);
path.SetPosition(2, -pathY);
path.SetPosition(3, -pathZ);


// Turn of path after target is found.

if (poseY <= 5 && poseY >= -5)
{
    count += 1;
}
if (count >= 65)
{
    pathHolder.SetActive(false);
}

// Depth bar
mSlider.value = poseY;
```

```
        // ****************Display text

        // Displat the relative pose as text, F1 declares the amount
           of decimales
        poseX_text.text = "X: " + Pose.x.ToString("F1");
        poseY_text.text = "Y: " + Pose.y.ToString("F1");
        poseZ_text.text = "Z: " + Pose.z.ToString("F1");

        // Display the relative pose as text on the cordsys
        kord_X.text = "X: " + Pose.x.ToString("F1");
        kord_Y.text = "Y: " + Pose.y.ToString("F1");
        kord_Z.text = "Z: " + Pose.z.ToString("F1");

        // For the improvement on the depth bar (Not used during the
           experiment)
        if (poseY < 0)
        {
            barfix.SetActive(true);
        }
        if(poseY > 0)
        {
            barfix.SetActive(false);
        }
        barY_text.text = "Y: " + Pose.y.ToString("F1");

        ////Water depth
       WaterDepth = -(poseZ - 1000);
       WaterDepthText.text = "Water depth: " +
          WaterDepth.ToString("F0") + " meters";
    }
}
```

Listing C.2: The C# script used in operation 1

Operation 2

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;


public class Path_illu : MonoBehaviour
{

    // Pose variables
    [SerializeField]
    private Transform marker;
    [SerializeField]
    private Transform ARCam;
    private float poseX;
    private float poseY;
    private float poseZ;
    private float pose;
    private float ToolPoseX;
    private float ToolPoseY;
    private float ToolPoseZ;
    private float TargetPoseX;
    private float TargetPoseY;
    private float TargetPoseZ;
    private float WaterDepth;
    [SerializeField]
    private GameObject operation2;
    [SerializeField]
    private GameObject tool2;
    private int count;
    [SerializeField]
    private GameObject info;


    // Depth bar
    public Slider CSlider;

    // Path
    public LineRenderer path;
    private Vector3 ToolEnd;
    [SerializeField]
    private GameObject toolMarker;
    //public int X;
    //public int Y;
    //public int Z;
    private float MarkerPoseX;
```

```
private float MarkerPoseY;
private float MarkerPoseZ;
private float EstX;
private float EstY;
private float EstZ;
private int i;
[SerializeField]
private GameObject cleaner;

// Path colors
public Color Green = Color.green;
public Color Red = Color.red;
public Color Blue = Color.blue;

//Path Filter
private List<float> PoseEstX = new List<float>() { 0,0,0,0,0 };
private List<float> PoseEstY = new List<float>() { 0,0,0,0,0 };
private List<float> PoseEstZ = new List<float>() { 0,0,0,0,0 };
private int NumerOfPoseEst;

// Text variables
[SerializeField]
private TextMesh poseX_text;
[SerializeField]
private TextMesh poseY_text;
[SerializeField]
private TextMesh poseZ_text;
[SerializeField]
private TextMesh kord_X;
[SerializeField]
private TextMesh kord_Y;
[SerializeField]
private TextMesh kord_Z;
[SerializeField]
private Text WaterDepthText;
public GameObject pathHolder;




// Start is called before the first frame update
void Start()
{

    Vector3 ARcamPose = new Vector3(0, 0, 0);

    // Configure the path
```

```
        path.startWidth = 0.35f;
        path.endWidth = 0.35f;

        // General UI
        operation2.SetActive(true);
        tool2.SetActive(true);

        count = 0;



    }

    // Update is called once per frame
    void Update()
    {


        /// Pose of the tool tip, with reference from the marker on
           the tool
        ToolPoseX = toolMarker.transform.position.x -13;
        ToolPoseY = toolMarker.transform.position.y;
        ToolPoseZ = toolMarker.transform.position.z;


        // Pose of tracking marker
        MarkerPoseX = marker.transform.position.x;
        MarkerPoseY = marker.transform.position.y +19;
        MarkerPoseZ = marker.transform.position.z +14;

        // Relative pose
        poseX = -(MarkerPoseX - ToolPoseX);
        poseY = -(MarkerPoseY - ToolPoseY);
        poseZ = -(MarkerPoseZ - ToolPoseZ);

        Vector3 Pose = new Vector3(poseX, poseY, poseZ);

        // Depth bar
        CSlider.value = Pose.y;


        // Add pose estimates to list
        PoseEstX.Add(poseX);
        PoseEstY.Add(poseY);
        PoseEstZ.Add(poseZ);

        i = PoseEstX.Count;
```

```
//    // Get the average of the 4th last poses
foreach (float value in PoseEstX)
{
    EstX = (PoseEstX[i - 4] + PoseEstX[i - 3] + PoseEstX[i -
       2] + PoseEstX[i - 1]) / 4;
    EstY = (PoseEstY[i - 4] + PoseEstY[i - 3] + PoseEstY[i -
       2] + PoseEstY[i - 1]) / 4;
    EstZ = (PoseEstZ[i - 4] + PoseEstZ[i - 3] + PoseEstZ[i -
       2] + PoseEstZ[i - 1]) / 4;

}


// Set the origin of the path at the tip of the tool
path.transform.position = new
   Vector3(toolMarker.transform.position.x - 18,
   toolMarker.transform.position.y,
   toolMarker.transform.position.z);

// Color change based on distance in y-direction from target

// Convert distance between tool and target to a 0f and 1f
   range
float lerp = Mathf.InverseLerp(0, 60, poseY);
float newinterval = Mathf.Lerp(0f, 1f, lerp);

// Lerp the color between far and near
Color lerpColor = Color.Lerp(Blue, Red, newinterval);
path.GetComponent<Renderer>().material.color = lerpColor;

// Turn off instructions
if(poseY <= 5 && poseY >= -5)
{
    count += 1;
}
if (count >= 200)
{
    info.SetActive(false);
}


//Clean
if (poseY <= 30)
{
    cleaner.SetActive(true);
}
```

```
    else
    {
        cleaner.SetActive(false);
    }

    // Path off
    if (poseY <= 5 && poseY >= -5)
    {
        count += 1;
    }
    if (count >= 200)
    {
        pathHolder.SetActive(false);
    }

    //Define the 3D line from the avrage of pose estimates for a
       more stable line representation.
    Vector3 pathX = new Vector3(0, EstY, 0);
    Vector3 pathY = new Vector3(0, EstY, EstZ);
    Vector3 pathZ = new Vector3(EstX, EstY, EstZ);


    // Assign the vector lines to the line renderer
    path.SetPosition(1, -pathX);
    path.SetPosition(2, -pathY);
    path.SetPosition(3, -pathZ);

    //Display text

    // Displat the relative pose as text, F1 declares the amount
       of decimales
    poseX_text.text = "X: " + Pose.x.ToString("F1");
    poseY_text.text = "Y: " + Pose.y.ToString("F1");
    poseZ_text.text = "Z: " + Pose.z.ToString("F1");

    // Display the relative pose as text on the cordsys
    kord_X.text = "X: " + Pose.x.ToString("F1");
    kord_Y.text = "Y: " + Pose.y.ToString("F1");
    kord_Z.text = "Z: " + Pose.z.ToString("F1");


    // Water depth
    WaterDepth = -(poseZ - 1000);
    WaterDepthText.text = "Water depth: " +
       WaterDepth.ToString("F0") + " meters";

}
```

```
}
```

Listing C.3: The C# script used in operation 2

Operation 3

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Operation3 : MonoBehaviour
{
    [SerializeField]
    private GameObject operation3;
    [SerializeField]
    private GameObject tool3;

    // Start is called before the first frame update
    void Start()
    {
        operation3.SetActive(true);
        tool3.SetActive(true);

    }
```

Listing C.4: The C# script used in operation 3

Main menu

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Menu : MonoBehaviour
{


    public void operation1()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           +2);
    }

    public void operation2()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           + 3);
    }

    public void operation3()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           + 4);
    }

    public void NonAR()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           + 1);
    }

    public void MainMenu_1()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           - 2);
    }

    public void MainMenu_2()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           - 3);
    }

    public void MainMenu_3()
```

```
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           - 4);
    }

    public void MainMenu_nAR()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
           - 1);
    }

    public void Quit ()
    {
        Application.Quit();
    }
}
```

Listing C.5: The C# script used to create the main menue

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Klokke : MonoBehaviour
{
    public Text ClockDisplayed;
    public Text DateDisplayed;

    // Start is called before the first frame update
    void Start()
    {
        //ClockDisplayed = GetComponent<Text>();
        //DateDisplayed = GetComponent<Text>();
    }

    // Update is called once per frame
    void Update()
    {
        // Get date
        System.DateTime date = System.DateTime.Today;

        DateDisplayed.text = date.ToString("MM.dd.yyyy");
        DateDisplayed.text = date.ToString("dd.MM.yyyy");

        // Get time
        System.DateTime time = System.DateTime.Now;

        string hour = time.Hour.ToString();
        string minute = time.Minute.ToString();
        string second = time.Second.ToString();

        ClockDisplayed.text = time.ToString(hour + ";" + minute + ";"
           + second);

    }
}
```

Listing C.6: The C# script used to enable the time and date function in the general UI

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Timer : MonoBehaviour
{
    public Text timerText;
    private float startTime;

    // Start is called before the first frame update
    void Start()
    {
        startTime = Time.time;
    }

    // Update is called once per frame
    void Update()
    {
        float t = Time.time - startTime;

        string minutes = ((int)t / 60).ToString("F0");
        string seconds = (t % 60).ToString("F0");

        timerText.text = "Operation time: " + minutes + ":" + seconds;
    }
}
```

Listing C.7: The C# script used to enable the operation timer function in the general UI

Appendix D

Python OpenCV script used for ArUco marker tracking

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 20 18:16:43 2019

@author: Alexa
"""

import numpy as np
import cv2
import cv2.aruco as aruco
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from mpl_toolkits import mplot3d




cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
marker_size = 4 # In [cm]

############################## Calibration Data
calib_path = ""
camera_matrix = np.loadtxt(calib_path+’Camera_calibrating_2.txt’,
   delimiter=’,’)
camera_distortion =
   np.loadtxt(calib_path+’Camera_calibrating_2_distortion.txt’,
   delimiter=’,’)
```

```
############################ Needed definitions
id_list = []
font = cv2.FONT_HERSHEY_SIMPLEX

############################  Initiation of plot
#ax = plt.axes(projection = '3d')

X = []
Y = []
Z = []


########################### Run the tracking loop
while (True):
   ######################## Initial Aruco code

    #Information grabed from the webcam
    ret, frame = cap.read()

    ### Operations on the camera frame (webcam feed) BGR -->Gray
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    ### Definition of marker library
    aruco_dict = aruco.Dictionary_get(aruco.DICT_6X6_250)
    parameters = aruco.DetectorParameters_create()

    #Lists of ids and the corners beloning to each id
    corners, ids, rejectedImgPoints = aruco.detectMarkers(gray,
       aruco_dict, parameters=parameters)

    ############################ Define what to do if marker is
       detected
    if np.all(ids != None):
        ider = ids.tolist()
        #print(ider)
        #### Estimate pose of each marker and return the values rvet
           and tvec---different from camera coefficients
        rvec, tvec ,_ = aruco.estimatePoseSingleMarkers(corners,
           marker_size, camera_matrix, camera_distortion)


        ## Multiple trackers position
        transform_200 = []
        transform_150 = []
```

```
###################seperate the incomming tvec information
   based on the corresponding marker for cube
n = 0
tag = []
for i in range(0, len(ider)):
    if ider[i][0] == 200:
        transform_200.append(tvec[n][0])
        tag.append(i)
        z = transform_200[i][2] + 3
    else:
        transform_150.append(tvec[n][0])
    n += 1
print(tag)

############################## Calculate ArUco cube center
if len(tag) == 1:
    x = ((corners[tag[0]][0][0][0] + corners[tag[0]][0][1][0]
       + corners[tag[0]][0][2][0] + corners[tag[0]][0][3][0])
       / 4)

    y = ((corners[tag[0]][0][0][1] + corners[tag[0]][0][1][1]
       + corners[tag[0]][0][2][1] + corners[tag[0]][0][3][1])
       / 4)
elif len(tag) == 2:
    x = ((corners[tag[0]][0][0][0] + corners[tag[0]][0][1][0]
       + corners[tag[0]][0][2][0] + corners[tag[0]][0][3][0]
    + corners[tag[1]][0][0][0] + corners[tag[1]][0][1][0] +
       corners[tag[1]][0][2][0] + corners[tag[1]][0][3][0] ) /
       8)

    y = ((corners[tag[0]][0][0][1] + corners[tag[0]][0][1][1]
       + corners[tag[0]][0][2][1] + corners[tag[0]][0][3][1]
    + corners[tag[1]][0][0][1] + corners[tag[1]][0][1][1] +
       corners[tag[1]][0][2][1] + corners[tag[1]][0][3][1] ) /
       8)

elif len(transform_200) == 3:
    x = ((corners[tag[0]][0][0][0] + corners[tag[0]][0][1][0]
       + corners[tag[0]][0][2][0] + corners[tag[0]][0][3][0]
    + corners[tag[1]][0][0][0] + corners[tag[1]][0][1][0] +
       corners[tag[1]][0][2][0] + corners[tag[1]][0][3][0]
    + corners[tag[2]][0][0][0] + corners[tag[2]][0][1][0] +
       corners[tag[2]][0][2][0] + corners[tag[2]][0][3][0]) /
       12)

    y = ((corners[tag[0]][0][0][1] + corners[tag[0]][0][1][1]
       + corners[tag[0]][0][2][1] + corners[tag[0]][0][3][1]
```

```
    + corners[tag[1]][0][0][1] + corners[tag[1]][0][1][1] +
       corners[tag[1]][0][2][1] + corners[tag[1]][0][3][1]
    + corners[tag[2]][0][0][1] + corners[tag[2]][0][1][1] +
       corners[tag[2]][0][2][1] + corners[tag[2]][0][3][1]) /
       12)


else:
    pass


############ draw circles at center
X_cent_2D = np.round(x).astype(np.int64)
Y_cent_2D = np.round(y).astype(np.int64)


cv2.circle(frame, (X_cent_2D,Y_cent_2D), 150 , (200,200,20), 5)
cv2.circle(frame, (X_cent_2D,Y_cent_2D), 5 , (200,200,20), 5)

############################# Make drawings
### Draw Axis of multiple markers
for i in range(0, ids.size):
    aruco.drawAxis(frame, camera_matrix, camera_distortion,
       rvec[i], tvec[i], 2)

### Draw the marker border
aruco.drawDetectedMarkers(frame, corners) #Draw A square
   around the markers

## If two markers are detected, print the following position
   information
if len(tag) >= 2:
    ## Pose marker 1
    str_position1 = ’MARKER1 Position x=%4.0f y=%4.0f
       z=%4.0f’%(x, y, z)
    cv2.putText(frame, str_position1, (100,100), font, 1,
       (0,0,255), 1, cv2.LINE_AA)
else:
    pass
for i in range(0,len(ider)):
    if ider[i][0] == 150:
        str_position2 = ’MARKER2 Position x=%4.0f y=%4.0f
           z=%4.0f’%(transform_150[0][0], transform_150[0][1],
           transform_150[0][2])
        cv2.putText(frame, str_position2, (100,200), font, 1,
           (0,0,255), 1, cv2.LINE_AA)
```

```
                relative_pose = 'Relative Position x=%4.0f y=%4.0f
                    z=%4.0f'%(x - transform_150[0][0], y -
                    transform_150[0][1],
                                                                          z
                                                                             -
                                                                             transfo


                cv2.putText(frame, relative_pose, (100,300), font, 1,
                    (0,0,255), 1, cv2.LINE_AA)
            else:
                pass

    # Display the resulting frame
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Listing D.1: Python OpenCV script for ArUco marker tracking

Bibliography

[1] Macrotrends. *WTI Crude Oil Prices - 10 Year Daily Chart*. 2019. URL: `https://www.macrotrends.net/2516/wti-crude-oil-prices-10-year-daily-chart` (visited on 05/23/2019).

[2] Equinor. *How we cut the break-even price from USD 100 to USD 27 per barrel*. 2018. URL: `https://www.equinor.com/no/magazine/achieving-lower-breakeven.html` (visited on 05/23/2019).

[3] Giorgio Biscardini; Reid Morrison; David Branson; Adrian Del Maestro. *Oil and Gas Trends 2018-19*. 2018. URL: `https://www.strategyand.pwc.com/trend/2018-oil-gas` (visited on 05/23/2019).

[4] Norwegian Petroleum Directorate. *The Shelf in 2018*. 2019. URL: `https://www.npd.no/en/facts/news/general-news/2019/The-Shelf-in-2018/` (visited on 05/23/2019).

[5] Aksel Andreas Transeth. *SEAVENTION*. 2018. URL: `https://www.sintef.no/en/projects/seavention/` (visited on 05/23/2019).

[6] Thomas J. Impelluso et al. *Modelling Subsea ROV robotics using the moving frame method*. 2018. URL: `https://link.springer.com/content/pdf/10.1007%5C%2Fs40435-018-0471-6.pdf` (visited on 05/23/2019).

[7] Fredrik Rødne Jenssen. *Dynamic Analysis of ROV Operation*. 2016. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2350695` (visited on 05/23/2019).

[8] Eirik Storesund. *Surface Assisted Autopilot for Remotely Operated Vehicle*. 2018. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2560559` (visited on 05/23/2019).

[9] Erik Hexeberg Henriksen. *ROV Control System for Positioning of Subsea Modules.* 2014. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/239244` (visited on 05/23/2019).

[10] Magnus Nørstebø. *ROV Intervention Tool Skid (RITS): Development of a new ROV tool.* 2014. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2452461` (visited on 05/23/2019).

[11] Aksel Andreas Transeth. *SEAVENTION –AUTONOMOUS SUBSEA INTERVENTION.* 2018. URL: `https://www.sut.org/wp-content/uploads/2018/05/015-Seavention-Autonomous-Subsea-Intervention-A-Transeth-Sintef.pdf` (visited on 05/23/2019).

[12] Alexander Næsheim. *Augmented Reality and Computer Vision for Subsea operations.* 2018. URL: `https://drive.google.com/file/d/1eFgkGdLJSf5c0fyJLpHnTcmP66a9H` `view?usp=sharing` (visited on 05/13/2019).

[13] Usability First. *Glossary. egocentric viewpoint.* 2019. URL: `http://www.usabilityfirst.com/glossary/egocentric-viewpoint/` (visited on 02/15/2019).

[14] Kristian T Simsarian and Karl-Petter Åkesson. *Windows on the World: An example of Augmented Virtuality.* 1999. URL: `https://www.researchgate.net/publication/2349157_Windows_on_the_World_An_example_of_Augmented` (visited on 02/15/2019).

[15] Louise Yarnall and Rakesh Kumar. *Augmented Reality.* 2015. URL: `http://sk.sagepub.com/reference/the-sage-encyclopedia-of-educational-technology/i1586.xml` (visited on 02/15/2019).

[16] Craig B Alan. "Understanding Augmented Reality : Concepts and Applications". In: Elsevier Science & Technology1, 2013. Chap. 1.

[17] 2019. URL: `https://www.digi-capital.com/news/2018/01/ubiquitous-90-billion-ar-to-dominate-focused-15-billion-vr-by-2022/` (visited on 05/19/2019).

[18] 2019. URL: `https://medium.com/ipg-media-lab/has-your-brand-been-augmented-yet-d4b7bab30e49` (visited on 05/19/2019).

[19] Kyle ROche. "iOs 5 Augmented Reality". In: 2011. Chap. 1.

[20] Snapchat inc. *Snapchat.* 2019. URL: `https://www.snapchat.com/l/nb-no/` (visited on 05/19/2019).

[21] D. Cushnan and H El Habbak. “UDeveloping AR games for iOS and Android : develop and deploy augmented reality apps using Vuforia SDK and Unity 3D”. In: 2013. Chap. 1.

[22] Stefanie Motte. *Augmented Reality: A Comprehensive History (Part 1)*. 2018. URL: `https://blog.vertebrae.com/history-augmented-reality-1` (visited on 05/19/2019).

[23] 2019. URL: `https://commons.wikimedia.org/wiki/File:Virtual-Fixtures-USAF-AR.jpg` (visited on 05/19/2019).

[24] Fabian Quint; Frieder Loch; Partick Bertram. *The Challenge of Introducing AR in Industry - Results of a Participative Process Involving Maintenance Engineers*. 2017. URL: `https://reader.elsevier.com/reader/sd/pii/S2351978917304687?token=24A4E16999F7A1A3424A442DF8472EA4BF2DF08CA9BD88984AEEB3139546F7C9` (visited on 05/31/2019).

[25] Holger Glockner; Kai Jannek; Johannes Mahn; Bjorn Theis. *AUGMENTED REALITY IN LOGISTICS*. 2014. URL: `https://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/csi_augmented_reality_report_290414.pdf` (visited on 05/31/2019).

[26] Volkswagen. *Virtual Technologies*. 2017. URL: `https://www.volkswagenag.com/en/group/research/virtual-technologies.html` (visited on 05/31/2019).

[27] Scope AR. *Know how, know now*. 2019. URL: `https://www.scopear.com/solutions/worklink-assist/` (visited on 05/31/2019).

[28] Volkswagen. 2019. URL: `https://www.volkswagen-vans.co.uk/en/about-us/news/augmented-reality.html` (visited on 05/31/2019).

[29] Andreas Hansen Lie and Øyvind Tjomsland. *Installation and IMR on a subsea production system*. 2019. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2452461` (visited on 05/22/2019).

[30] 2019. URL: `https://www.moog.com/products/multiplexers-media-converters/focal-multiplexer-product-line/subsea-oilfield-communications-multiplexers.html` (visited on 05/31/2019).

[31] UTC Norway. *Bergen – the subsea capital of the world*. 2019. URL: `https://www.utc.no/utf/om_subsea/the_subsea_industry_in_western_norway/` (visited on 05/22/2019).

[32] Equinor. *Invisible platforms*. 2019. URL: `https://www.equinor.com/en/magazine/the-final-frontier.html` (visited on 05/22/2019).

[33] NORSOK. *NORSOK Z-008:2017*. 2019. URL: `https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=956000` (visited on 05/22/2019).

[34] Rigzone. *How do ROVs work?* 2008. URL: `https://www.rigzone.com/training/insight.asp?insight_id=343&c_id=` (visited on 05/22/2019).

[35] 2019. URL: `https://www.oceaneering.com/datasheets/ROV-Millennium-Plus.pdf` (visited on 05/22/2019).

[36] Schjølberg et al. *Next Generation Subsea Inspection, Maintenance and Repair Operation*. 2016. URL: `https://bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=TN_sciversesciencedirect_elsevierS2405-8963(16)32031-6&context=PC&vid=NTNU_UB&search_scope=default_scope&tab=default_tab&lang=no_NO` (visited on 04/22/2019).

[37] 2019. URL: `https://www.innova.no/e/products/rov-tooling-products/manipulator-systems/schilling-robotics-master-controller/` (visited on 05/22/2019).

[38] 2019. URL: `https://www.lerus-training.com/blog/offshore-operations/rov-certification-and-roles-for-personnel/` (visited on 05/22/2019).

[39] Usability.gov. *Usability Evaluation Basics*. 2017. URL: `https://www.usability.gov/what-and-why/usability-evaluation.htmll` (visited on 05/09/2019).

[40] Tom Stewart. *Usability evaluation*. 2009. URL: `https://www.tandfonline.com/doi/full/10.1080/01449290902786510?scroll=top&needAccess=true` (visited on 05/09/2019).

[41] INTERNATIONAL STANDARD. *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems (ISO 9241-210:2010)*. 2010. URL: `https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=466417` (visited on 04/10/2019).

[42] Rogers et al. "INTERACTION DESIGN beyond human-computer interaction". In: John Wiley and Sons, 2011. Chap. 1.

[43] Julie A. Adams. *Critical Considerations for Human-Robot Interface Development*. 2002. URL: `https://www.cs.rit.edu/~jaa/Papers/AAAISymposium.pdf` (visited on 04/11/2019).

[44] Mica R. Endsley. *DESIGN AND EVALUATION FOR SITUATION AWARENESS ENHANCEMENT*. 1988. URL: `https://journals.sagepub.com/doi/pdf/`

10.1177/154193128803200221?casa_token=gMXgLkvPFAYAAAAA:RBn7Uc4eZ2iTqDO
(visited on 04/11/2019).

[45] CDFM-A. Hodge Leah. *Decision Support.* 2007. URL: https://search.proquest.com/docview/194759583?rfr_id=info%5C%3Axri%5C%2Fsid%5C%3Aprimo (visited on 04/11/2019).

[46] Øyvind Netland. *Remote Inspection of Offshore Wind Turbines: study of the Benefits, Usability and Feasibility.* 2014. URL: https://brage.bibsys.no/xmlui/handle/11250/261141 (visited on 02/15/2019).

[47] Anirudha Joshi and N.L. Sarda. *Do Teams Achieve Usability Goals? Evaluating Goal Achievement with Usability Goals Setting Tool.* 2011. URL: https://link.springer.com/content/pdf/10.1007%5C%2F978-3-642-23774-4.pdf (visited on 05/07/2019).

[48] at Michigan Tech University Robert Pastel. *CS4760 & CS5760: Human-Computer Interactions & Usability.* 2018. URL: http://cs4760.csl.mtu.edu/2017/lectures/usability/ (visited on 04/13/2019).

[49] Based on: Human-computer interaction by Alan Dix et al. *Usability: Goals, Guidelines and Principles.* 2004. URL: http://users.cs.cf.ac.uk/A.I.Abdelmoty/Teaching%5C%20material/W5-%5C%20233-Usability%5C%20Principles.pdf (visited on 04/13/2019).

[50] Lars Kolbeinstveit. *Hva er effektivitet?* 2019. URL: https://www.civita.no/politisk-ordbok/hva-er-effektivitet (visited on 04/10/2019).

[51] Craig Donald. "People in control : human factors in control room design". In: London: Institution of Electrical Engineers., 2001. Chap. 3.

[52] E.J. Sanders M.S. & McCormick. "Human Factors in Engineering and Design. Industrial Robot: An International Journal, p.153". In: New York: McGraw-Hill, 1998.

[53] Marcia W. Blenko and Michael C. Mankins. *Measuring decision effectiveness.* 2012. URL: https://www.bain.com/insights/measuring-decision-effectiveness/ (visited on 04/14/2019).

[54] Usability.gov. *Usability Evaluation Basics.* 2019. URL: https://www.usability.gov/what-and-why/usability-evaluation.html (visited on 05/02/2019).

[55] Whitney Quesenbery. "Usability in Government Systems". In: 2012 Elsevier Inc., 2012. Chap. 20.

[56] John Brooke. *SUS - A quick and dirty usability scale*. 1996. URL: https://hell.meiert.org/core/pdf/sus.pdf (visited on 03/10/2019).

[57] Brooke Falcone Mandy R. Drew and Wendy L. Baccus. *What Does the System Usability Scale (SUS) Measure? Page 356-366*. 2018. URL: https://link.springer.com/content/pdf/10.1007%5C%2F978-3-319-91797-9.pdf (visited on 05/02/2019).

[58] U.S. Department of Health & Human Services. *System Usability Scale (SUS)*. 2018. URL: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html (visited on 05/02/2019).

[59] PhD Jeff Sauro. *MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS)*. 2011. URL: https://measuringu.com/sus/ (visited on 04/25/2019).

[60] James R. Lewis and Jeff Sauro. *The Factor Structure of the System Usability Scale. Page 94-103*. 2009. URL: https://link.springer.com/content/pdf/10.1007%5C%2F978-3-642-02806-9.pdf (visited on 04/29/2019).

[61] Aida Meredassa Kumssa. *Tablet User Interface Evaluation for a Portable Ultrasound System and Real time Doppler SpectrumProcessing*. 2013. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/261076 (visited on 03/19/2019).

[62] James Miller Aaron Bangor Philip Kortum. *An Empirical Evaluation of the System Usability Scale*. 2008. URL: https://www.tandfonline.com/doi/citedby/10.1080/10447310802205776?scroll=top&needAccess=true (visited on 04/28/2019).

[63] James Miller Aaron Bangor Philip Kortum. *Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*. 2009. URL: http://uxpajournal.org/wp-content/uploads/sites/8/pdf/JUS_Bangor_May2009.pdf (visited on 04/28/2019).

[64] Microsoft. *Install the tools*. 2019. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools (visited on 03/11/2019).

[65] Unity. *Visual Studio C# integration*. 2018. URL: https://docs.unity3d.com/Manual/VisualStudioIntegration.html (visited on 03/11/2019).

[66] Wikipedia. *Vuforia Augmented Reality SDK*. 2019. URL: https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK (visited on 05/19/2019).

[67] Vuforia. *Getting Started with Vuforia Engine in Unity.* 2019. URL: `https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.htmll` (visited on 03/11/2019).

[68] PTC Vuforia. *Optimizing Target Detection and Tracking Stability.* 2019. URL: `https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability` (visited on 02/22/2019).

[69] PTC Vuforia. *Image Targets.* 2019. URL: `https://library.vuforia.com/articles/Training/Image-Target-Guide` (visited on 05/20/2019).

[70] LLC SmartDraw. *Selecting the Right Software Development Tools for Your Developers.* 2019. URL: `https://www.smartdraw.com/technology/right-software-development-tools-for-developers.htm` (visited on 01/20/2019).

[71] Lentin Joseph. *Learning robotics using python.* 2015. URL: `https://bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=BIBSYS_ILS71520400920002201&context=L&vid=NTNU_UB&search_scope=default_scope&tab=default_tab&lang=no_NO` (visited on 01/20/2019).

[72] Unify Community Wiki. *Running unity on Linux thorugh Wine.* 2019. URL: `https://wiki.unity3d.com/index.php/Running_Unity_on_Linux_through_Wine` (visited on 01/21/2019).

[73] SaveDan. *Can I run two webcams simultaneously on one computer?* 2011. URL: `https://www.makeuseof.com/answers/run-webcams-simultaneously-computer/` (visited on 01/25/2019).

[74] Ivideon. *How to use a webcam in two applications at the same time?* 2019. URL: `https://www.ivideon.com/help/start/using-a-webcam-in-two-applications-at-the-same-time/` (visited on 01/25/2019).

[75] ROS Wiki. *cv bridge Tutorials.* 2019. URL: `http://wiki.ros.org/cv_bridge/Tutorials/UsingCvBridgeToConvertBetweenROSImagesAndOpenCVImages` (visited on 01/20/2019).

[76] ROS Wiki. *Creating a workspace for catkin.* 2019. URL: `http://wiki.ros.org/catkin/Tutorials/create_a_workspace` (visited on 01/18/2019).

[77] ROS Wiki. *ROS Tutorials.* 2019. URL: `http://wiki.ros.org/ROS/Tutorials` (visited on 01/15/2019).

[78] Enrique Fernandez Fernández & Fernández. *Learning ROS for robotics programming : your one-stop guide to the Robot Operating System.* 2015. URL: `https:`

//bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=BIBSYS_ILS71520531660002201&context=L&vid=NTNU_UB&search_scope=default_scope&tab=default_tab&lang=no_NO (visited on 01/15/2019).

[79] Nathan Jennings. *Socket Programming in Python (Guide)*. 2018. URL: https://realpython.com/python-sockets/ (visited on 01/30/2019).

[80] Tiziano Fiorenzani. *How do drones work*. 2019. URL: https://github.com/tizianofiorenzani/how_do_drones_work/tree/master/opencv (visited on 01/20/2019).

[81] Stack Overflow community. *Unity: Live Video Streaming*. 2017. URL: https://stackoverflow.com/questions/42717713/unity-live-video-streaming (visited on 02/05/2019).

[82] Gur Raunaq Singh. *Introduction to Using OpenCV With Unity*. 2018. URL: https://www.raywenderlich.com/5475-introduction-to-using-opencv-with-unity (visited on 02/03/2019).

[83] Thomas Mountainborn. *UNITY AND OPENCV – PART ONE: INSTALL*. 2016. URL: https://thomasmountainborn.com/2016/09/11/unity-and-opencv-part-one-install/ (visited on 02/08/2019).

[84] Thomas Mountainborn. *UNITY AND OPENCV – PART TWO: PROJECT SETUP*. 2016. URL: https://thomasmountainborn.com/2016/09/12/unity-and-opencv-part-two-project-setup/ (visited on 02/08/2019).

[85] ENOX SOFTWARE. *OpenCV for Unity*. 2019. URL: https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088 (visited on 01/20/2019).

[86] Stack overflow community. *Has anyone used aruco with unity for marker tracking?* 2017. URL: https://stackoverflow.com/questions/42009324/has-anyone-used-aruco-with-unity-for-marker-tracking (visited on 02/08/2019).

[87] PTC Vuforia. *Vuforia developer database*. 2019. URL: https://developer.vuforia.com/vui/develop/databases (visited on 02/20/2019).

[88] Stack overflow community. *How does Vuforia image recognition work?* 2019. URL: https://stackoverflow.com/questions/12253664/how-does-vuforia-image-recognition-work (visited on 04/21/2019).

[89] Quora community. *What algorithm does Vuforia use to track images and extract the features of images?* 2015. URL: `https://www.quora.com/What-algorithm-does-Vuforia-use-to-track-images-and-extract-the-features-of-images` (visited on 04/21/2019).

[90] PTC Vuforia. *VuMark.* 2019. URL: `https://library.vuforia.com/articles/Training/VuMark` (visited on 02/25/2019).

[91] QRCode-Generator.de. *QR Code Generator.* 2019. URL: `https://www.qr-code-generator.com/` (visited on 03/01/2019).

[92] Usability.gov. *User Interface Design Basics.* 2017. URL: `https://www.usability.gov/what-and-why/user-interface-design.html` (visited on 03/02/2019).

[93] PTC Creo. *Creo Parametric 3D Modeling Software.* 2019. URL: `https://www.ptc.com/en/products/cad/creo/parametric` (visited on 04/21/2019).

[94] Sergey Chernenko. *Mean filter, or average filter.* 2010. URL: `http://www.librow.com/articles/article-5` (visited on 03/20/2019).

[95] Unity. *Color.Lerp.* 2019. URL: `https://docs.unity3d.com/ScriptReference/Color.Lerp.html` (visited on 05/12/2019).

[96] Jakob Nielsen. *Why You Only Need to Test with 5 Users.* 2000. URL: `https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/` (visited on 03/10/2019).

[97] Jakob Nielsen and Thomas K. Landauer. *A Mathematical Model of the Finding of Usability Problems.* 1993. URL: `http://peres.rihmlab.org/Classes/PSYC6419seminar/p206-Five%5C%20Users%5C%20nielsen.pdf` (visited on 03/10/2019).

[98] Norwegian offshore rental. *ROV Tools.* 2019. URL: `https://offshorerental.no/products/rov-tools/` (visited on 04/30/2019).

[99] 2019. URL: `https://helpx.adobe.com/acrobat/using/playing-video-audio-multimedia-formats.html` (visited on 06/02/2019).

[100] Logitech. *LOGITECH C925E.* 2019. URL: `https://www.logitech.com/en-us/product/c925e-webcam` (visited on 04/30/2019).

[101] CFPHS Chauntelle Baughman. *Hot Stabs: Quick Couplings on Steroids.* 2015. URL: `https://www.hydraulicspneumatics.com/fittings-couplings/hot-stabs-quick-couplings-steroids` (visited on 03/10/2019).

[102] Fugro. *Fugro DeepData ROV Deployment*. 2015. URL: `https://www.youtube.com/watch?v=U19O_-pfB6w` (visited on 04/22/2019).

[103] MADS SOEGAARD. *Usability: A part of the User Experience*. 2019. URL: `https://www.interaction-design.org/literature/article/usability-a-part-of-the-user-experience` (visited on 04/30/2019).

[104] Google. *Create beautiful forms*. 2019. URL: `https://www.google.com/intl/en-GB/forms/about/` (visited on 04/30/2019).

[105] Human Performance Research Group NASA Ames Research Center. *TASK LOADINDEX*. 1986. URL: `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000021488.pdf` (visited on 05/16/2019).

[106] Norsk standard. *Remotely operated vehicle (ROV) services*. 2013. URL: `https://www.standard.no/pagefiles/978/u-102r1.pdf` (visited on 02/25/2019).

[107] Catherine Diaz et al. *Designing for Depth Perceptions in Augmented Reality*. 2017. URL: `http://www.danszafir.com/papers/ISMAR17-Szafir.pdf` (visited on 02/28/2019).

[108] National Science Foundation Miles O'brien. *Improving our depth perception in augmented reality*. 2012. URL: `https://phys.org/news/2012-09-depth-perception-augmented-reality.html` (visited on 04/28/2019).

[109] Jungsik Park et al. *Binocular mobile augmented reality based on stereo camera tracking*. 2016. URL: `https://core.ac.uk/download/pdf/81710467.pdf` (visited on 04/28/2019).

[110] Vuforia. *Getting Started with Vuforia Engine in Unity*. 2017. URL: `https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html` (visited on 05/12/2019).

[111] Erkan Bostanci. *Motion model transitions in GPS-IMU sensor fusion for user tracking in augmented reality*. 2015. URL: `https://arxiv.org/pdf/1512.02758.pdf` (visited on 05/28/2019).

[112] Schön F. Gustafsson and Per Slycke. *Sensor Fusion for Augmented Reality*. 2006. URL: `https://www.researchgate.net/publication/224281289_Sensor_Fusion_for_Augmented_Reality` (visited on 05/28/2019).

[113] Abir Khan. *Deep Reinforcement Learning based tracking behavior for underwater vehicles*. 2018. URL: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2564506` (visited on 03/06/2019).

[114] J. Lucas Lucchetta. *6 Pillars of UI Design*. 2016. URL: `https://blog.prototypr.io/6-pillars-of-ui-design-80fe255f5091` (visited on 05/11/2019).

[115] Germaine Satia. *Familiarity in User Experience*. 2015. URL: `http://infinvision.com/familiarity-in-user-experience/` (visited on 05/11/2019).

[116] Usability first. *Familiarity*. 2019. URL: `http://www.usabilityfirst.com/glossary/familiarity/` (visited on 05/11/2019).

[117] Argon design. *5 aspects of a good user interface*. 2014. URL: `http://www.argondesign.com/news/2014/feb/5/5-aspects-good-user-interface/` (visited on 05/13/2019).

[118] Microsoft. *Microsoft Excel*. 2019. URL: `https://products.office.com/en/excel` (visited on 05/04/2019).

[119] Gougoux et al. *A functional neuroimaging study of sound localization: visual cortex activity predicts performance in early-blind individuals*. 2005. URL: `https://www.ncbi.nlm.nih.gov/pubmed/15678166` (visited on 05/11/2019).

[120] John Staughton. *Does Losing One Sense Improve the Others?* 2016. URL: `https://www.scienceabc.com/humans/does-losing-one-sense-improve-the-others.html` (visited on 05/11/2019).

[121] Google. *Google sheets*. 2019. URL: `https://www.google.com/sheets/about/` (visited on 05/04/2019).

[122] R.N. et al. Smith. *Adaptive Path Planning for Tracking Ocean Fronts with an Autonomous Underwater Vehicle*. 2016. URL: `https://link.springer.com/chapter/10.1007%5C%2F978-3-319-23778-7_50` (visited on 05/13/2019).

[123] Wang Hong-Jian ; Zhao Jie ; Bian Xin-Qian ; Shi Xiao-Cheng. *An improved path planner based on adaptive genetic algorithm for autonomous underwater vehicle*. 2005. URL: `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1626663` (visited on 05/13/2019).

[124] Zeng Jiangfeng et al. *Adaptive line-of-sight path following control for underactuated autonomous underwater vehicles in the presence of ocean currents*. 2017. URL: `https://journals.sagepub.com/doi/full/10.1177/1729881417748127` (visited on 05/13/2019).

[125] Jens Keil; Florian Schmitt; Timo Engelke; Holger Graf; Manuel Olbrich. *Augmented Reality Views: Discussing the Utility of Visual Elements by Mediation Means in Industrial AR from a Design Perspective, page 298-312*. 2018. URL: `https://link.`

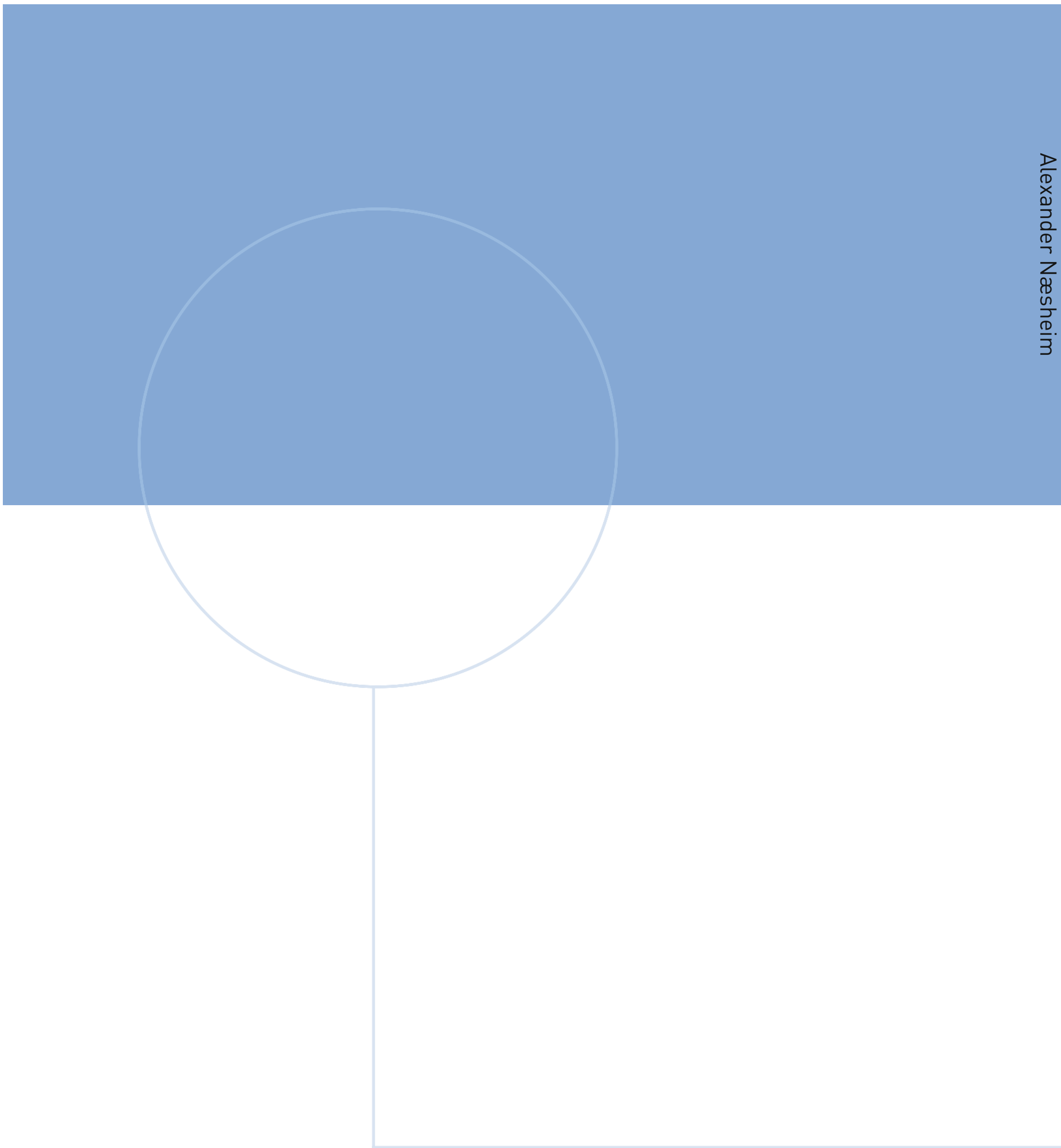
`springer.com/content/pdf/10.1007%5C%2F978-3-319-91584-5.pdf` (visited on 05/28/2019).

[126] *Rov pilots*. 2019. URL: `http://ajan.ciceros.co/rov-pilots/` (visited on 05/14/2019).

[127] WILLIAM H. DELONE and EPHRAIM R. MCLEAN. *The DeLone and McLean Model of Information Systems Success: A Ten-Year Update*. 2003. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.3031&rep=rep1&type=pdf` (visited on 05/15/2019).

[128] Natalie Skinner et al. *Workplace Support*. 2005. URL: `http://nceta.flinders.edu.au/files/5812/4710/5736/Workplace_Support.pdf` (visited on 05/19/2019).

[129] TrymyUI. *SUS: The System Usability Scale*. 2019. URL: `https://www.trymyui.com/sus-system-usability-scale` (visited on 05/06/2019).



Alexander Næsheim

NTNU
Kunnskap for en bedre verden