Utilizing Hydra for Real-Time Reconstruction of Environments in Extended Reality

Hannes Rydén Sonesson and Edward Sjöblom

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH — LUND UNIVERSITY

2024

MASTER THESIS

Utilizing Hydra for Real-Time Reconstruction of Environments in Extended Reality

Hannes Rydén Sonesson
`ha0351so-s@student.lu.se`

Edward Sjöblom
`ed2576sj-s@student.lu.se`

January 23, 2024

Master thesis work carried out at Ericsson AB.

Supervisors: Joakim Eriksson, `joakim.eriksson@design.lth.se`
Héctor Caltenco, `hector.caltenco@ericsson.com`

Examiner: Günter Alce, `gunter.alce@design.lth.se`

Utilizing Hydra for Real-Time Reconstruction of Environments in Extended Reality

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

Subject: Interaction Design MAMM01
Division: Ergonomics and Aerosol Technology
Supervisor: Joakim Eriksson
Examiner: Günter Alce

Abstract

This thesis explores the use of Hydra for real-time reconstruction of environments in extended reality (XR).

The development of the prototype was performed in an iterative manner. Three iterations were executed, each resulting in a prototype with improvements related to the takeaways from the preceding prototype. The final prototype consisted of two scenes. The first scene consisted of rendered furniture according to the Hydra output from a pre-recorded ROS bag file of an office scene. The second scene consisted of the same room definitions, but was empty of furniture. As Hydra ran, the furniture was rendered in real-time. After the third iteration, user tests were performed in order to evaluate the prototype. The results of the user tests indicated that there was potential of utilizing Hydra, but some areas were identified for further improvements.

Lastly, the discussion described the code complexity and configuration challenges of Hydra, highlighting the need for documentation and systematic configuration testing. Bottlenecks in the workflow, such as object handling and message structure, suggest some techniques for optimizing the data pipeline. The potential for custom input and appropriately configuring Hydra emerges as interesting areas for further exploration.

Keywords: extended reality, usability testing

Sammanfattning

I denna rapport undersöks hur Hydra kan användas för återupbyggnad av miljöer i extended reality (XR).

Utvecklingen av prototypen genomfördes iterativt. Tre iterationer genomfördes där varje iteration resulterade i en prototyp med förbättringar utifrån den föregående prototypen. Den slutgiltiga prototypen bestod av två scener. Den första scenen bestod av statiskt renderade möbler enligt data från Hydra. Den andra scenen hade samma definitioner av rummet som den första, men var tom på möbler. När Hydra kördes renderades möblerna i realtid. Användartester genomfördes efter den tredje iterationen för att utvärdera den resulterande prototypen. Resultaten från användartesterna indikerade att det fanns potential hos Hydra för återuppbyggnad av miljöer i realtid. Några potentiella förbättringsområden identifierades.

Slutligen diskuterades kodens komplexitet i Hydra samt de utmaningar som uppstod vid konfigureringen av Hydra. Behovet av dokumentation och systematisk testning av konfigurationen belystes. Flaskhalsar i dataflödet beskrevs och tekniker för att optimera dataflödet föreslogs. Potentialen för egeninspelad inmatning samt konfigurering av Hydra beskrevs som intressanta områden för framtida studier.

Nyckelord: extended reality, användbarhetstestning

Acknowledgements

We would like to thank our supervisor Joakim Eriksson at the Department of Design Sciences at LTH for his continuous feedback and support during the thesis. We would also like to thank our supervisors Héctor Caltenco, Hiba Alqaysi and Leire Amenabar at Ericsson for their continuous support, feedback and valuable discussions regarding the thesis work. We would also like to thank the Ericsson research employees and the others who participated in the user tests, providing us with valuable insights.

Acronyms and abbreviations

- **XR** - Extended Reality
- **AR** - Augumented Reality
- **VR** - Virtual Reality
- **HMD** - Head-Mounted Display
- **3DSG** - 3D Scene Graph
- **ROS** - Robot Operating System
- **SLAM** - Simultaneous Localization and Mapping

Contents

Chapter 1
Introduction

1.1 Background

The development of applications and hardware for Extended Reality (XR) is on the rise. New XR hardware is released continuously, pushing the boundaries of the technology. This indicates that XR could be a more integrated and natural part of our lives in the future [2].

The workflow of scanning a real environment and rendering it in real-time in XR marks a significant leap in merging physical objects with virtual objects. Utilizing techniques as photogrammetry, LiDAR, depth sensors and machine learning, spatial details may be defined and applied as a basis for reconstructing a virtual environment from a real environment [15]. Once defined, the challenge of rendering the virtual objects in an XR environment emerges, demanding computational power to process the data while ensuring a low latency, delivering a satisfying user experience [2].

A 3D scene graph (3DSG) is a high-level representation of a 3D environment. It is described as a layered graph, consisting of nodes that represent spatial concepts at multiple levels of abstraction. A framework that constructs 3D scene graphs from sensor data in real-time is Hydra [13]. Based on simulated data, Hydra is able to build 3D scene graphs with an accuracy similar to batch offline methods despite running online [13]. Utilizing Hydra, the workflow of scanning a real environment and rendering it in real-time in XR may be possible.

The thesis was written in collaboration with Ericsson. Ericsson was founded in 1876 and has over the past 140 years been a global leader in ICT solutions. The company manufactured some of the world's first telephones, then managing networks processing 40% of the world's data. Continuous innovation is a key area of Ericsson, driving innovation in telecommunications, connectivity, sustainability, and digital transformation across industries [9].

1.2 Purpose and Research Questions

The purpose of the thesis is to explore the role of Hydra and 3DSG while reconstructing a virtual environment from a real environment in real-time, as a user experiences the virtual environment in XR. Additionally, by examining a specific use case from a user-centered design perspective, the ambition is to provide knowledge regarding spatial perception in a virtual environment, where virtual objects resemble real objects.

RQ1: How is a user's spatial awareness affected in a virtual environment reconstructed from a real environment with Hydra?

RQ2: How can real objects be tracked and rendered in real-time in the virtual environment?

RQ3: How does the user perceive the virtual environment while objects are rendered in real-time?

RQ4: What is the use of 3DSG and Hydra in an XR context?

1.3 Scope and Limitations

The work of the thesis had some restrictions of scope and limitations that was set to meet the time restriction of the thesis work of approximately 20 weeks. The scope and limitations should be considered when interpreting the result, discussion and conclusion that follows.

Configuring Hydra posed challenges that could have been improved with an extended time period of the work. As Hydra lacked documentation to a quite large extent, it was not always clear how modifications to Hydra affected the output. Hence, some arbitrary choices of configuration were implemented by testing different configurations in an exploratory manner, visually analyzing the output of Hydra.

Additionally, generating custom input to Hydra with a camera was not implemented due to the time restriction of the thesis and lack of knowledge regarding the area. The initial ambition was to set it up, however, during the work of the thesis it was decided to shift focus to other parts of the thesis. As the generation of custom input to Hydra was not implemented, pre-recorded input to Hydra was utilized during the user tests. Accordingly, tracking and rendering of real objects in real-time was not performed during the user tests.

1.4 The Sustainable Development Goals

The Sustainable Development Goals were created by United Nations in 2015 to serve as a blueprint for peace and prosperity for people and the planet. In total, 17 goals were defined [18].

The work of the thesis can primarily be connected to goal number 9: *Industry, Innovation* *and Infrastructure*. As XR can be considered a progressive and innovative technology, the ambition of the thesis to improve a workflow within XR aligns well with the goal. Additionally, goal number 11: *Sustainable Cities and Communities* may to some degree be connected to the work of the thesis. As the rapid growth of cities is happening, tools for urban planning may be in demand. Scanning a real environment and visualizing it as a virtual environment may be of use in order to perform urban planning [18].

Figure 1.1: Goal number 9.

Figure 1.2: Goal number 11.

1.5 Related Work

Although there has been a vast amount of research done regarding reconstructing virtual environments from real environments, few have focused on executing it in real-time as a user is experiencing the virtual environment in XR.

In the article *Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions*, Çöltekin et. al. investigates the role of XR in the spatial context. One finding is that processing a scan of a real object to a virtual object in real-time in XR demands significant computational power [2].

A literature review of real-time 3D reconstruction techniques was presented by Ingale and Udayan. The authors argue that despite the significant improvements of the reconstruction of scene dynamics that have taken place in recent years, the existing techniques are considerably limited. Self-occlusion of objects is handled poorly and topological changes in real-time is yet a significant challenge [15].

Hughes et. al. present Hydra that enables the construction of 3D scene graphs from sensor data in real-time, creating a persistent representation of the environment [13]. In a following article, Hughes et. al. highlight that Hydra can run in real-time on embedded computers used for robotics in real-time, without the need for a powerful workstation. It is also concluded that the choice of 2D semantic network may improve the object detection [14].

A research team at Ericsson worked on converting the Aria Digital Twin dataset [19] to function as input to Hydra. The procedure could provide a foundation in the work of generating input to Hydra with a camera recording in real-time.

By combining areas from the related work, a foundation for the development of the workflow during the thesis was set. Çöltekin et. al. provided understanding about the computational power of rendering the environment in XR in real-time. Thus, the C# scripts for generating objects was written with an ambition to reduce the time complexity in order to improve the performance in Unity. Ingale and Udayan provided understanding about the potential limitations of real-time reconstruction of environments to be aware of. The Hydra framework that Hughes et. al. presented served as a basis for generating a virtual representation of a 3D environment that the generated objects in Unity was based on. Lastly, the work with the Aria Digital Twin dataset by the research team at Ericsson served as basis regarding getting the dataset to run with Hydra.

Chapter 2
Technical Background

2.1 Extended Reality

Extended reality (XR) is an umbrella term that includes Virtual reality (VR) and Augmented reality (AR). It is a spectrum of technologies that combine physical and virtual worlds to varying degrees of immersion and interactivity. VR is used with the ambition to entirely immerse the user in virtual worlds, while AR is used to overlay virtual information on the physical world [1].

2.2 Computer Vision

Computer vision is a field of computer science focusing on enabling computers to interpret and understand visual information from images and videos. Researchers in computer vision have developed mathematical techniques to replicate human capabilites such as image recognition, object detection, scene reconstruction and motion analysis. During the last two decades, the progress has been rapid. There are reliable techniques for computing a 3D model of an object using overlapping 2D images of an object [29].

2.3 Programming Languages

2.3.1 C#

C# is an object oriented and general purpose programming language created by Microsoft. It was first released in 2002, but has evolved much since then. Its effectiveness comes from been inspired as a blend between C and C++. C# runs on Microsofts .NET framework and common language runtime (CLR) which is a virtual execution system. There are a lot of different use cases but are mainly used in backend services, windows application, website development and game development [3, 33]. The game engine Unity is one example of where it can be used for game logic, by defining desired functionality in one or more script and adding it to a GameObjects [32].

2.3.2 C++

C++ is a high-level general purpose programming language created by Bjarne Stroustrup. It was released as an extension of the C programming language in 1985, but it has since then evolved substantially [28].

2.3.3 Python

Python is an object-oriented high-level general purpose programming language created by Guido van Rossum in 1991. Although it is defined as an object-oriented programming language, it also supports multiple programming paradigms such as functional and procedural programming. It has a simple syntax, and it comes with a large standard library that covers many different areas of programming such as; string processing, internet protocols, operating system interfaces and software engineering. There are also an extensive variety of well-documented libraries that can be utilized for specific purposes [21].

2.4 Unity

Unity is a versatile game engine that can be used to develop 3D game applications. It includes libraries and tool kits for XR development, providing the user with tools to improve the XR development process [31].

2.5 Windows Subsystem for Linux

Windows Subsystem for Linux (WSL) is a tool developed by Microsoft which allows users to run a Linux environment on Windows operating systems [4].

2.6 ROS

Robot Operating System (ROS) is an open-source framework providing a structured communications layer that operates above the host operating system within a heterogeneous compute cluster. ROS consists of four fundamental concepts; nodes, messages, topics and services. Nodes are deployed to perform computations. A system typically consist of many nodes, each node handling a specific task. Messages are defined as a strictly typed data structure, and they are passed between nodes in order to communicate. Nodes can publish messages to given topics. A node that is interested in a certain type of data subscribes to specific topic transferring that type of data. Multiple nodes can publish and subscribe concurrently to one topic, and one node can publish and subscribe to multiple topics. A service is defined by a pair of messages, one for a request and for a reply. A client can call a service by sending a request message and then await a reply. ROS bag files can be created to store message data by subscribing to topics. A recorded ROS bag file can be replayed [22].

ROS includes commands such as [24]:

- roscore - initialize the ROS Master, allowing nodes discover each other and establish communication channels.
- rosbag - perform various operation on ROS bag files such as, recording and playing.
- rosrun - run an executable in an arbitrary package.
- roslaunch - launch a set of nodes from an XML configuration file.
- rosnode - display node information.
- rosmsg - display the data structure definition of a message.
- rostopic - display information about a topic and print the messages sent on a topic.
- rossrv - display the data structure definition of a service.
- rqt_graph - display the current graph of nodes and topics.

2.7 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is the concept of deploying a robot at an unknown location in an unknown environment, letting the robot explore the unknown environment, progressively constructing an accurate map of the environment while simultaneously keeping track of its position in the map [8].

2.8 Hydra

Hydra is a real-time spatial perception system that builds and optimizes a 3D scene graph (3DSG) from sensor data (visual-inertial data) in real-time. The system is developed by MIT SPARK Lab with the aim of environmental understanding for robotics. Hydra takes a robot's sensor data and gradually builds 5 layers of a 3DSG, which is then constantly updated as the robot explores the environment simultaneously [13].

2.8.1 Architecture

The Hydra architecture consists of three modules across various processing rates. From real-time to slower intervals, ensuring efficient organization of tasks without hindering faster processes. Visualized in Figure 2.1, Hydra employs different modules for early, mid-level, and high-level perception tasks.

Starting with rapid early perception processes (left side of Figure 2.1), Hydra handles low-level tasks like feature tracking, 2D semantic segmentation, and stereo-depth reconstruction. These outputs feed into mid-level perception processes (center), constructing layers such as the agent, mesh/places, and object layers, alongside an initial scene graph. The high-level perception processes include loop closure detection, scene graph optimization and room detection. The result is a globally consistent 3D scene graph [14].

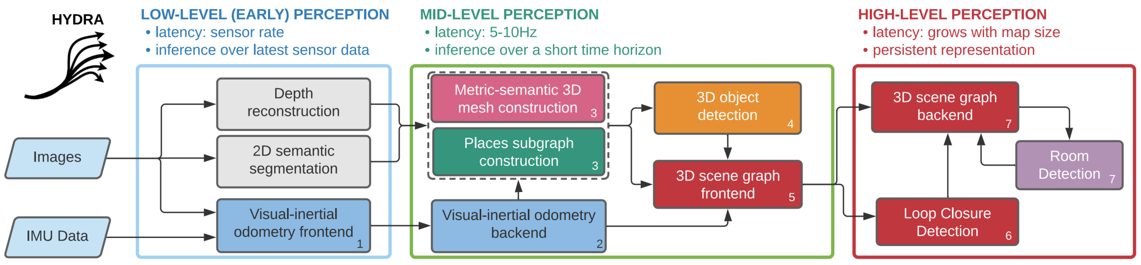

HYDRA
LOW-LEVEL (EARLY) PERCEPTION
• latency: sensor rate
• inference over latest sensor data
MID-LEVEL PERCEPTION
• latency: 5-10Hz
• inference over a short time horizon
HIGH-LEVEL PERCEPTION
• latency: grows with map size
• persistent representation
Images
IMU Data
Depth reconstruction
2D semantic segmentation
Visual-inertial odometry frontend 1
Metric-semantic 3D mesh construction 3
Places subgraph construction 3
Visual-inertial odometry backend 2
3D object detection 4
3D scene graph frontend 5
3D scene graph backend 7
Room Detection 7
Loop Closure Detection 6

Figure 2.1: Architecture of Hydra [13].

2.8.2 3D Scene Graph

A 3D scene graph is a high-level representation of a 3D environment. The 3D environment is described as a layered graph where spatial concepts are located at multiple levels of abstraction in the graph. Edges are used to describe relations between the spatial concepts.

When Hydra runs, it generates a 3DSG in real-time. The 3DSG consists of five layers:

- Layer 5 - Buildings
- Layer 4 - Rooms
- Layer 3 - Places
- Layer 2 - Objects and agents
- Layer 1 - Metric-semantic 3D mesh

Layer 1 consists of a metric-semantic 3D mesh. Layer 2 consists of a subgraph of objects and agent. Every object has a semantic label, a bounding box and a centroid. Every agent has a position and a graph describing its trajectory. Layer 3 consists of a subgraph of places where each place is a location without obstacles. Layer 4 consists of a subgraph of rooms. Each room has a centroid and edges to adjacent rooms, if adjacent rooms exist. Layer 5 is a node that connects all rooms into a building [14]. An illustration of a 3DSG generated by Hydra can be seen in Figure 2.2.

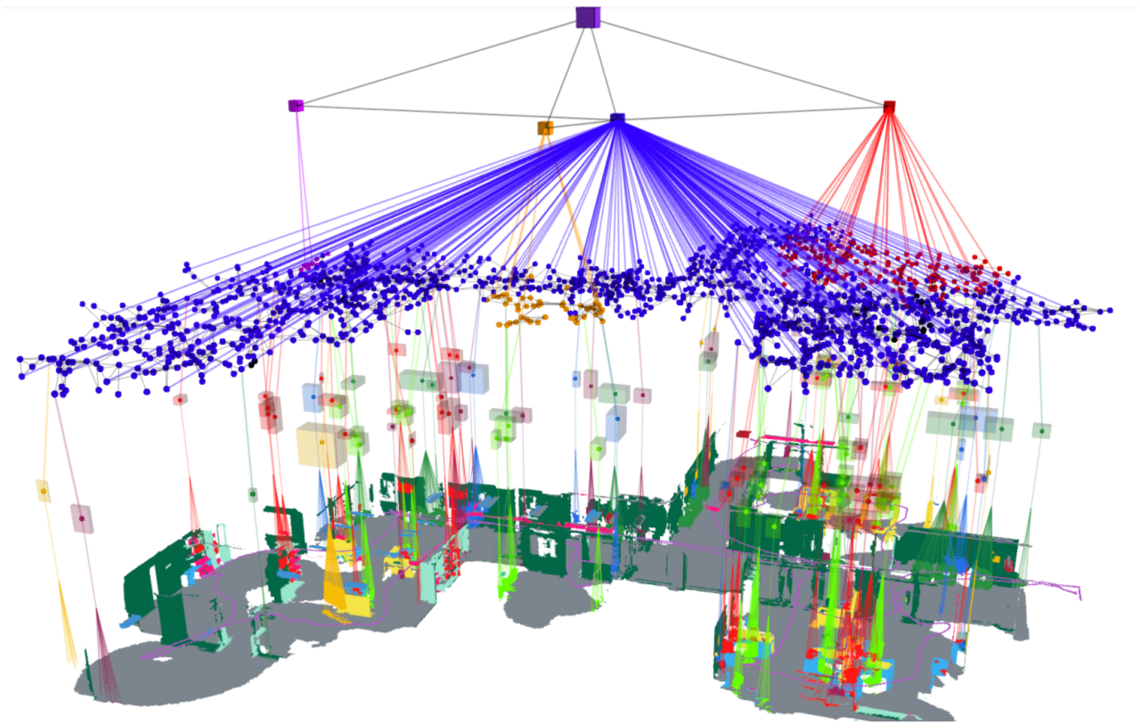


Figure 2.2: A 3DSG generated by Hydra [13].

2.8.3 Object Detection

Hydra performs its object detection from the 3D metric-semantic mesh vertices, created in layer 1, using Euclidean clustering. The sets of vertices are clustered based on their semantic labels. The individual clusters can then be used to create an estimated centroid and bounding box that represents an object. This process continues through the process and new objects are created every update. When a new object is detected it can either belong to an existing object node or be the start of a new one. In the first case the vertices of the new node is merged to the existing node and otherwise it is added as a new object node. A new node is determined to belong to an existing overlap and have the same semantic labeling, i.e. if the centroid is enclosed in the others bounding box.

2.8.4 Loop Closure Detection

Hydra uses a hierarchical approach for loop closure detection, visualized in Figure 2.3. In the first step, a top-down loop closure detection with hierarchical descriptors, capturing statistics from the layers in the 3DSG, is utilized in order to detect potential loop closures. In the second step, a bottom-up geometric verification is used to approximate the loop closure by detecting potential matches. In the third step, an algorithm is deployed to optimize the 3DSG from the data of the loop closure detection [14].

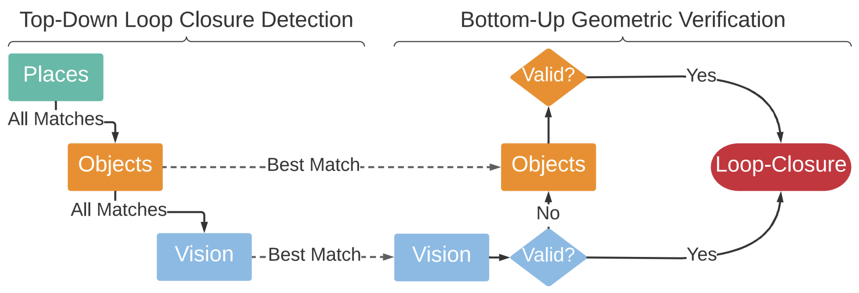


Figure 2.3: Loop closure detection of Hydra [13].

2.8.5 Semantic Labeling

For each key frame of the recorded data from the camera, a 2D semantic segmentation network is employed in order to label RGB image pixels and reconstruct depth maps using stereo matching or depth sensor data. The semantic segmentation is then combined with depth data from the camera in order to generate a semantic 3D point cloud, which is transformed according to the estimated position of the camera [14].

2.9 Kimera

Kimera is an open-source library written in C++ that performs real-time metric-semantic visual-inertial Simultaneous Localization and Mapping (SLAM). Kimera takes stereo images (camera feed) and IMU data as input and produces state estimation for the sensor (see Figure 2.4 (a)) and different metric-semantic mesh reconstructions of the environment (see Figure 2.4 (b-e)) by utilizing visual-inertial sensing [25]. Hydra uses Kimera in order to construct layer 1 and layer 2 of the 3DSG.

The library consists of four modules that are responsible for different parts of the process to provide customization options for the user. The modules consist of [25]:

- Kimera-VIO - uses visual-inertial odometry (VIO) to produce fast and precise state estimation.
- Kimera-RPGO - performs robust pose graph optimization (RPGO) to improve localization.
- Kimera-Mesher - efficiently produces a 3D mesh of the environment. Which is achieved from single- and multi-frame regularization.
- Kimera-Semantics - also produces a 3D mesh but with higher precision and slower performance, with a volumetric method. Furthermore it executes 2D pixel-wise semantic segmentation which semantically annotates the mesh.

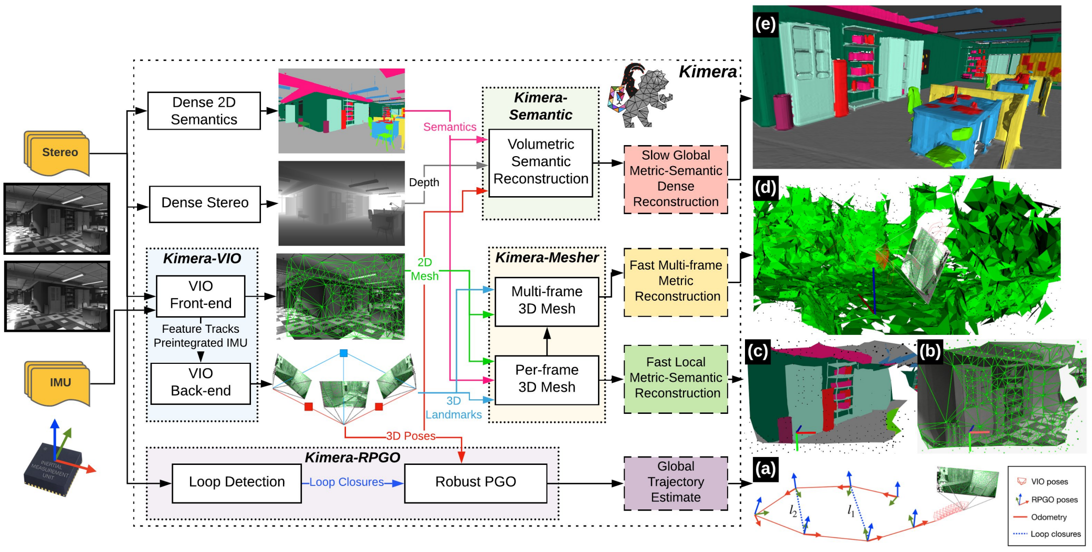


Figure 2.4: Architecture of Kimera [25].

2.10 Datasets

2.10.1 uHumans2

The uHumans2 dataset is a simulated Unity-based dataset that consists of three scenes: an office, an apartment and a subway station. The dataset has 2D semantic segmentation data, depth data, visual-inertial data and ground truth data of the robot trajectory that records the environment [13]. An image from the dataset can be seen in Figure 2.5.

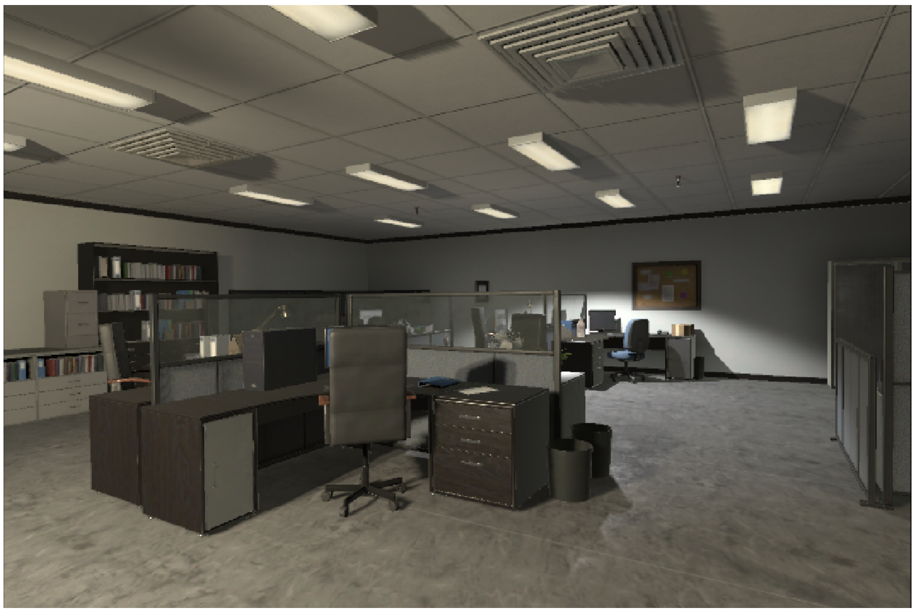


Figure 2.5: RGB image from the office scene in the uHumans2 dataset.

2.10.2 Aria Digital Twin

The Aria Digital Twin dataset is a dataset captured using Aria glasses [16]. The dataset includes 200 scenes. Each scene consists of data from two monochrome camera streams, data from one RGB camera stream, data from two IMU streams, complete sensor calibration, ground truth data of 6-degree-of-freedom (6DoF) poses of the Aria devices, object 6DoF poses, 3D eye gaze vectors, 3D human poses, 2D image segmentations, image depth maps and photo-realistic synthetic renderings [19]. An image from the dataset can be seen in Figure 2.6.

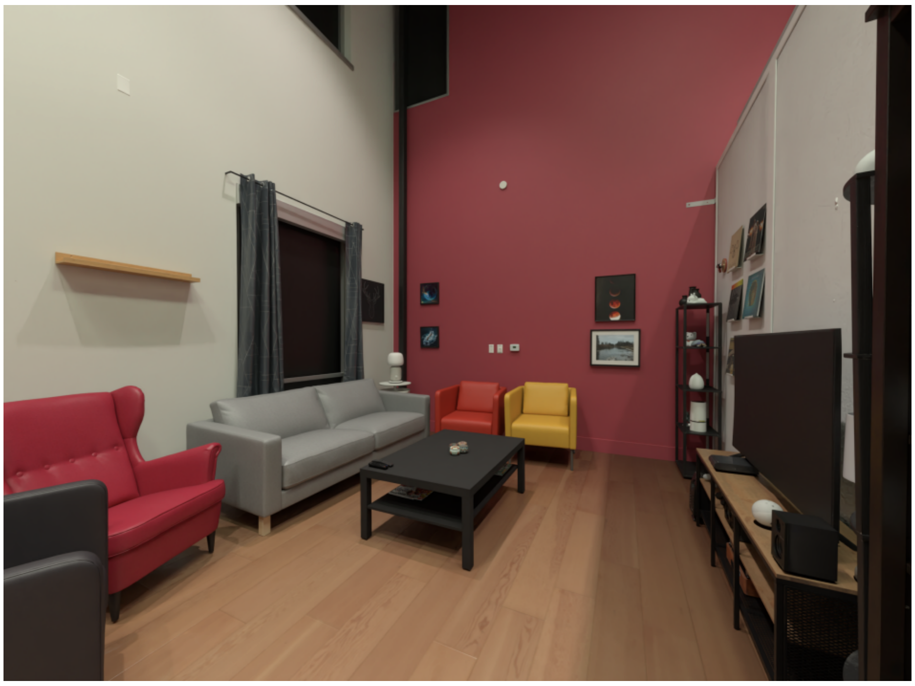


Figure 2.6: RGB image from the Aria Digital Twin dataset [19].

2.11 ROS TCP Connector

ROS TCP Connector is a tool developed by Unity Technologies that is used to facilitate communication between ROS and Unity. It establishes a TCP/IP connection, acting as a bridge between ROS and Unity. Accordingly, data can be exchanged between ROS nodes and Unity applications [30].

2.12 Devices

2.12.1 Meta Quest 2

Meta Quest 2 is a standalone VR headset developed by Meta. The headset has two displays, one for each eye, with a 1832 x 1920 resolution per display. It has built-in sensors and cameras enabling inside-out tracking, letting users experience virtual environments without the need for external sensors. The headset also includes two hand controllers [17].

2.12.2 Intel RealSense Depth Camera D435i

The Intel RealSense Depth Camera D435i is a depth camera which is a part of the RealSense D400 series developed by Intel. The camera is equipped with depth, RGB camera and inertial measurement unit (IMU). IMU provides movement and rotation in 6 degrees of freedom (6DoF) [23].

Chapter 3
Theoretical Background

3.1 Design Process

There are four main parts involved in the interaction design process; **establishing requirements**, **designing alternatives**, **prototyping** and **evaluating**. Furthermore it is an iterative process where each activity serve as the basis for the next one. The design alternatives should be based on the previously formulated requirements [20].

3.1.1 Double Diamond

The Design Council [7] developed in 2005 a design process model called the double diamond. The name is a reference to the shape that the four phases creates when visualizing the mindset that should be used during the different stages, see Figure 3.1. The two mindsets they refer to is diverging and converging, i.e. a stage is either convergent or divergent. The four phases are **discover**, **define**, **develop** and **deliver** [5, 27]. However the Design Council acknowledge that the design process is not a linear process and differs between projects. It might in other words be necessary to be flexible as new insights and changing conditions emerge during the the different stages [6].

Discover

The discovery is a divergent phase meaning it is used to broaden the perspective to generate a wide range of ideas for the product. In this stage it is important to gather information by getting an understanding of the product components, identifying requirements, problems and needs from users and stakeholders. The focus can differ depending on the project and the stakeholders, but typically includes: a user study, a literature study and market research. It is important to understand that this phase can in practice be ongoing during the entire process, due to new information and insights appearing during the process [5, 27].

Define

The define stage is a convergent stage where the end product should be clearly defined with problems to solve and with a plan to follow. It is achieved by taking all ideas, problems and hypotheses from the discovery phase, converting them to an action plan. There are multiple approaches, but some activities include: initial idea generation and general project development in order to limit the scope based on time limits and technological limitations [5, 27].

Develop

The development phase has similarities with the define phase but with a focus on the specific concepts defined. It is a divergent stage that often includes iteration in order to resolve the problems from the previous stages. The end goal is to deliver a product as close to the end product as possible. Some activities performed during the iterations are, ideation, visualization, development methods and testing. The idea behind visualization is to communicate the progress of the process, both within the development group and to stakeholders and users. Prototyping is commonly used as a development method to implement the ideas which are then used for the final activity: testing. Tests are important for multiple reasons. To ensure that usability requirements are fulfilled and improved from a previous iteration, to establish that predefined problems are addressed and to investigate if new problems or insights are discovered [5, 27].

Deliver

The deliver phase is a convergent phase where the final product is delivered. It involves taking the output from the development phase, ensuring that all problems are addressed from the discovery phase. It is also important that the acquired knowledge from the process is documented for the future. This stage includes final tests and evaluation [5, 27].

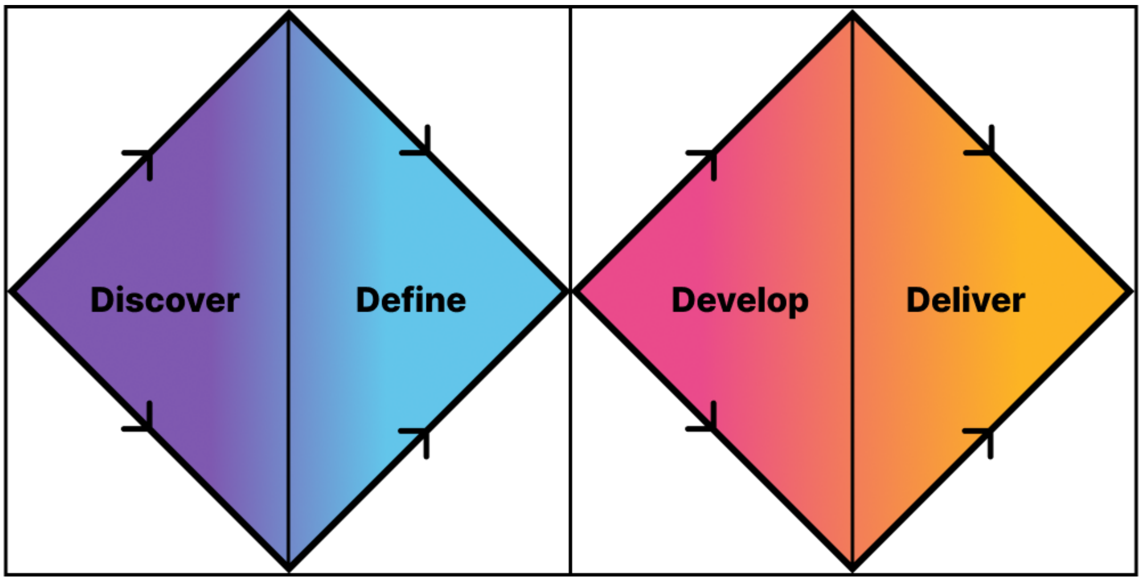


Figure 3.1: The double diamond design process model.

3.2 Usability Evaluation

3.2.1 Usability Testing

The overall goal of usability testing is to gather data to identify usability qualities and deficiencies in a product prior to releasing it. The ambition is to create products that are useful and valuable to the users, are easy to learn, enables effectiveness and efficiency and that are satisfying to use [26].

Test Plan

A test plan is written as the comprehensive blueprint that outlines the approach of testing the usability of a product. It serves as a way to concretize everything related to the testing that needs to be done. It is also an approach that establishes an understanding and agreement between the members that are involved in the testing [26].

Pilot test

A pilot test is planned and executed before the actual testing in order to ensure that the material of the test is clear and understandable, that all important tasks of the test is included and that the hardware, software, scripts and documentation are in order and ready for testing. Timing tasks during the pilot test can also be done to determine the time some pre-defined tasks will take in practice [26].

Selection of Participants

The selection of participants of a test is important since the participants should be representative of the intended users of the product that is tested. Selecting participants involves defining the relevant knowledge and skills of the users. Once it has been defined, a way to acquire participants for the test needs to be determined [26].

Test Moderator

The role of the test moderator is to have the ultimate responsible for all preparations for the test, including scripts, test equipment, participant arrangements and coordination of the other roles of the test team. The test moderator needs to have good communication skills and be a good coordinator, even though unexpected events may occurr during the test [26].

Data Gatherer

The role of the data gatherer is to register and classify data from expected critical activities and test events into categories. The data gatherer can keep track of the time of different tasks, write down answers to interview questions, handle audio and video recording and take notes on general observations during the test. The tools used to gather data during the test is determine before the test [26].

Informed Consent

An informed consent form is a written text that explains the study and describes the risks to the participant. It also emphasizes who the test person can contact with questions of the study [26].

Screening Questionnaire

A screening questionnaire is created with the purpose of finding a qualified selection of participants to the test. It can consist of questions regarding personal characteristics, requirements and background [26].

Orientation script

An orientation script is a written text that is read to the participants during the test. The script describes the background of the test, what will happen during the test, sets a tone for the participants for the test, with the intention of making the participants comfortable and at ease. It also reinforces the important statement that the it is the product that is being tested, not the participant.

Debriefing

A debriefing is held after a participant has done some tasks of a test. The ambition of a debriefing is to explore and review the participants actions and emotions during those tasks. While the performance of the tasks during the test can uncover problems of the tested product, it is commonly the debriefing session that highlights the aspects that led to the problems and that identifies how the participant experienced it [26].

Data Collection

The purpose of data collection is to gather all the important data that comes out of the test. The ambition is to collect data as simply, concisely and reliably as possible. There are many different tools that can be used for data collection, and the tools chosen should relate back to the objectives of the test and the research questions of the study [26].

3.2.2 Affinity Diagram

Affinity diagram is a technique used for analyzing quantitative data. Creating an affinity diagram consists of dividing ideas and answers into groups that share likeness or have the same theme. The goal is to get a better understanding of the data, establishing a structured overview. It is important to note that there are no predetermined groups or categories, they emerge during the process of creating the affinity diagram [10, 20].

The process of creating the Affinity diagram is usually performed by a group of people. It starts off by adding the data (ideas, observations, answers, etc) to individual notes. A note is then chosen at random as a starting point of a new group. The unsorted notes are looked through for similarities to the randomly chosen one, and the ones that are found are added to the group. This iterative process is continued until all notes belong to a group, with each iteration involving the creation of a new group based on the selection of a note from the unsorted ones [10, 20].

Chapter 4

Iteration 1

This chapter describes in detail the first iteration of the project process, including the stages involved and the specific activities for each stage. The designing and the development of the first prototype was based on the double diamond model, with the modification that the deliver phase was omitted, as seen in Figure 4.1. Lastly, some final takeaways of the iteration is presented.

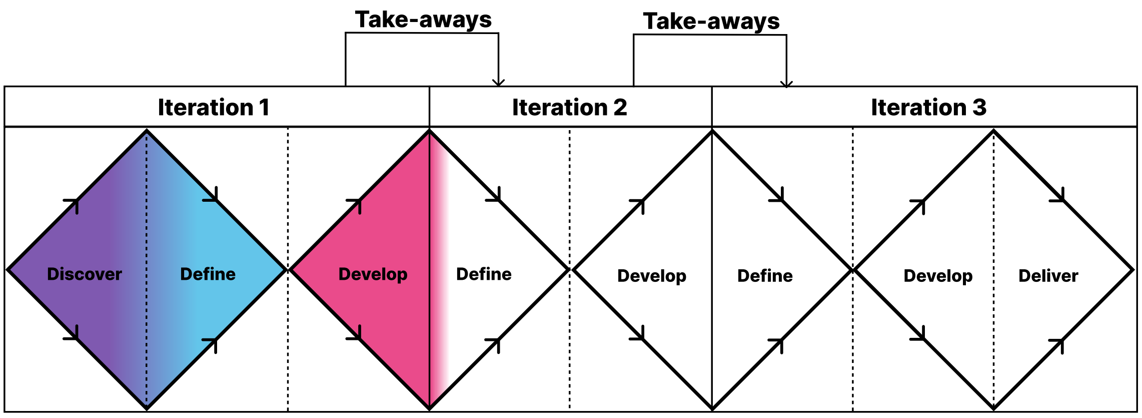


Figure 4.1: The stages of development involved in iteration 1.

4.1 Discover

The first part of the design phase consisted of discovering ideas for the product. Brainstorming sessions were held in order to come up with potential ideas to develop. Literature and papers regarding reconstruction of real environments to virtual environments were researched and read.

Ultimately, the Hydra framework was decided to be utilized as a basis for the development of the first iteration. It was chosen because of its claimed capabilities of generating 3D scene graphs in real-time from a scanned environment and because of its accuracy of the 3D scene graphs. The office scene from the uHumans2 dataset was used as input to Hydra during all iterations of development.

4.2 Define

The second part of the design phase consisted of defining a concept to strive for during the development. As the ultimate goal of the development was to provide a workflow that enabled scanning of a real environment and rendering it in XR in real-time, parts of the workflow could be split into more specific areas. For scanning the environment a camera was needed. In order to transfer the 3D environment data from Hydra to Unity via ROS, a data pipeline needed to be implemented. To reconstruct the 3D environment in Unity, Unity scripts needed to be implemented that rebuilt the environment according to the 3D environment data from the data pipeline. Collision detection to separate and merge objects in Unity was to be implemented.

The design phase lead to the definition of an ideal concept to strive for in the development, a concept including:

- A camera for scanning the environment.
- A Hydra-ROS-Unity data pipeline to transfer the 3D environment data from Hydra to Unity.
- Unity scripts for generating the environment according to the Hydra data.
- Collision detection of objects in Unity.
- An XR integration in Unity to let the user experience the environment being rendered in real-time using a HMD.

The aim was to put these parts together, building a full workflow of scanning an environment and rendering the environment, with the objects in the environment, in XR in real-time. An illustration of the concept can be seen in Figure 4.2.

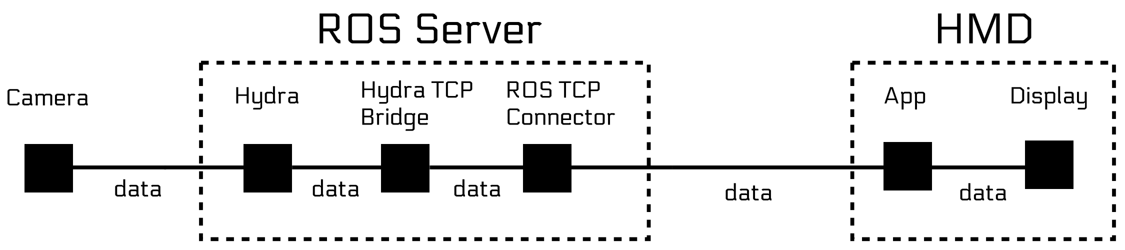


Figure 4.2: Overview of the defined concept.

4.3 Develop

In the initial stage of development phase a project plan was written. The development of the prototype was split into smaller tasks and a deadline for the tasks was set.

The first task was to install the software required for developing and running the prototype. First, Unity 2022.3.10f1 was installed. Then, WSL was installed. Further, ROS Noetic was installed. A ROS workspace was created. As a ROS workspace had been created, Hydra could be installed in the workspace.

Following, the next step was to get ROS to work with Unity in order to send data from Hydra via ROS to Unity. To achieve this data pipeline, ROS TCP Connector was installed.

The next step was to create a ROS node that kept track of the position of the camera in Hydra. The node subscribed to a topic where the positional data was sent. The node then published the data to a ROS node running in Unity. As the node in Unity received positional data, the scene was updated with the new position of the camera. The camera was in Unity represented by a sphere that had a purple trail to visualize its trajectory of movement.

As the Hydra-ROS-Unity data pipeline was in place, the reconstruction of objects in Unity according to the bounding boxes from Hydra could be implemented. First, a ROS node was created that subscribed to a ROS topic from Hydra where data about the positions, scaling and labeling of the bounding boxes was sent. A customized ROS message was then created. The message encapsulated positional, volumetric and label data for a specific bounding box. A ROS node was then created in Unity so that it could subscribe to a topic where the custom message could be sent. A Unity script was written to generate objects according to the specifications of the custom message. Each time the ROS node received data about a bounding box from Hydra the ROS node created a custom message with the corresponding data, and published it to the topic that the ROS node in Unity subscribed to. The Unity script could then generate the object according to the bounding box data of the custom message.

An illustration of the data pipeline with the ROS nodes can be seen in Figure 4.3, where the dotted line encapsulates the ROS nodes. The Hydra node was the ROS node started when running Hydra. The Hydra TCP Bridge consisted of two nodes: the ROS node subscribing to the positional data from the camera in Hydra and the ROS node subscribing to the ROS topic in Hydra where bounding box data was sent. The ROS TCP Connector was the ROS node enabling communication with Unity. The App was the ROS node that subscribed to the data transmitted over the ROS TCP Connector, in order to get information about the camera position and the bounding boxes of the objects from Hydra. As input to Hydra, the office scene from the uHumans2 data set was used.

All parts of the workflow was then put together in order to deliver a prototype. The outcome was a sphere in Unity that was moving around according to the positional data from Hydra, leaving a purple trail of its trajectory. As Hydra defined a bounding box of an object, the specifications of the bounding box was sent to Unity and an object was generated in Unity according to the bounding box specifications. The delivered prototype consisted of:

- A Hydra-ROS-Unity data pipeline to transfer the agent's position from Hydra to Unity in place.
- A Hydra-ROS-Unity data pipeline to transfer the objects' positions from Hydra to Unity in place.
- Unity script for visualizing the position of the agent the environment according to the Hydra data.
- Unity script for reconstructing objects based on their position in the environment according to the Hydra data.
- Unity script for collision detection of objects.

An overview of the implemented parts can be seen in Figure 4.3. An image from the Unity application can be seen in Figure 4.4.

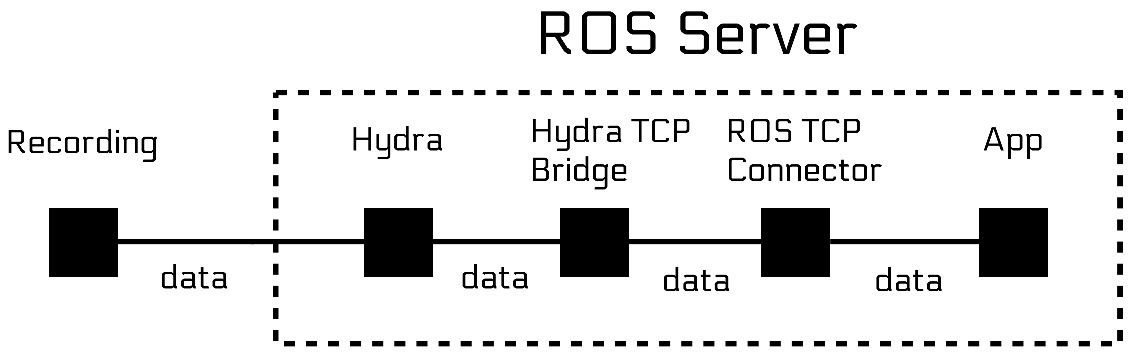


Figure 4.3: Overview of the implemented workflow.

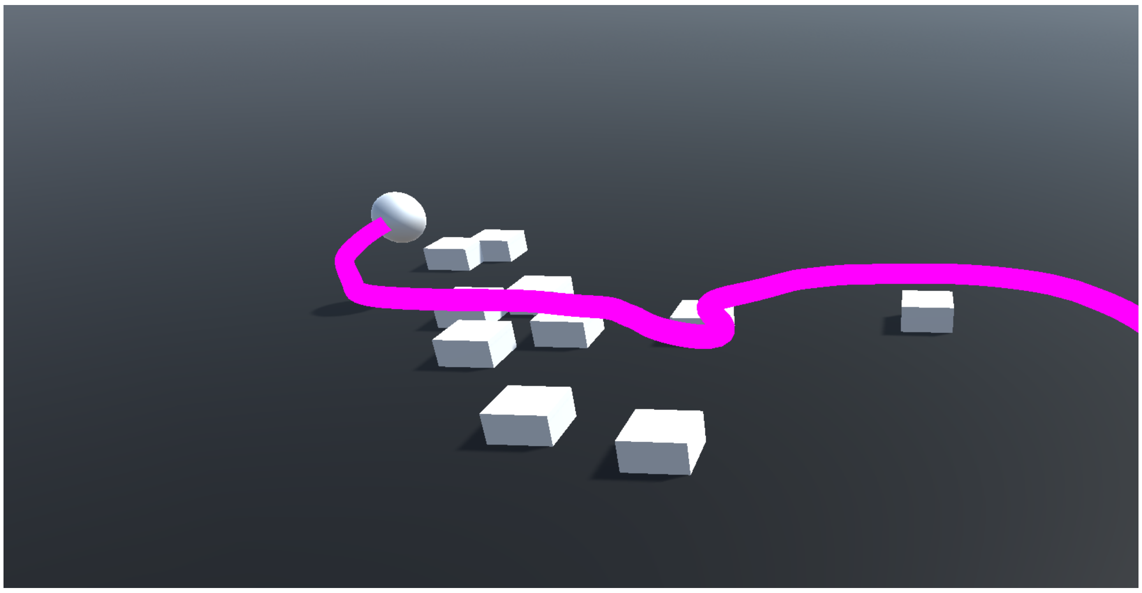


Figure 4.4: Unity application after the first iteration. The sphere illustrates the agent, with its trajectory visualized by the purple trail. The white blocks illustrates the detected objects.

4.4 Takeaways

After developing the first iteration of the workflow some takeaways to consider for the following iteration of the development was written. The takeaways were:

- The data pipeline Hydra-ROS-Unity was in place without noticeable dropouts. Accordingly, the basis for reconstructing a 3D environment in Unity based on data from Hydra was in place.
- Generating self-recorded input with a camera to Hydra was a task that was postponed to the next iteration of development due to lack of time to implement it during the first iteration.
- The labeling of the objects in Hydra was incorrect but consistent. The root of the problem was not identified and it was decided to further investigate it during the next iteration of development.
- By visualizing the bounding boxes with Hydra, it was clear that some bounding boxes incorrectly separated and merged certain objects. Accordingly, the bounding boxes of the objects in Hydra was identified as an area that could be focused on during the next iteration of development in order to improve the scaling, orientation and the merging and separation of objects that were located close to each other.
- The objects generated in Unity could be improved as they were represented by simple white blocks.

Chapter 5

Iteration 2

This chapter describes in detail the second iteration of the project process, including the stages involved and the specific activities for each stage. The designing and the development of the second prototype was based on the double diamond model, with the modification that the discover phase and the deliver phase was omitted, as seen in Figure 5.1. Lastly, some final takeaways of the iteration is presented.

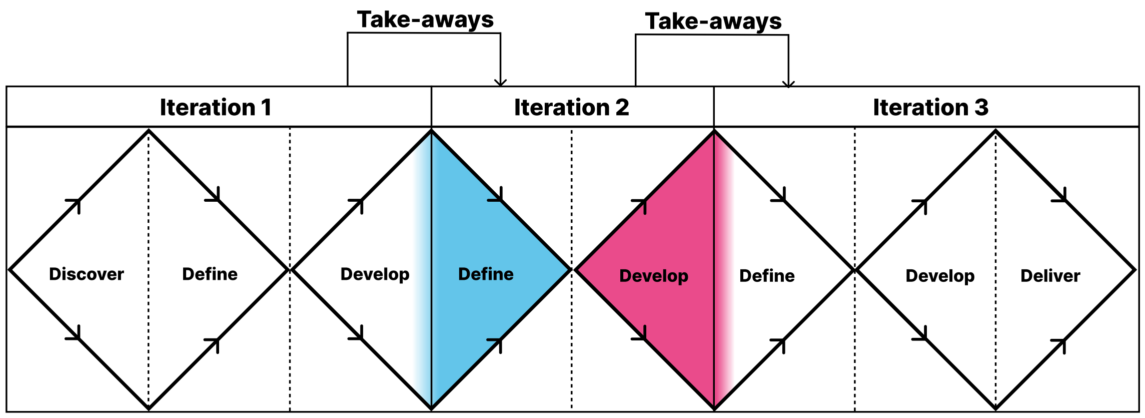


Figure 5.1: The stages of development involved in iteration 2.

5.1 Define

The first prototype and the takeaways for developing the first prototype were taken into account when defining the concept of the second prototype. As the data pipeline Hydra-ROS-Unity was in place with the visualization in Unity of the agent's position and the objects' positions from Hydra, the natural next step was to implement a more accurate reconstruction of the objects. The camera used to generate self-recorded input to Hydra was also defined as a task to focus on as it had not been implemented during the first iteration. Implementing correct labeling of the objects in Hydra was also defined as a task to focus on.

The concept to strive for when developing the second iteration of the prototype was defined as:

- Replace the Unity objects with 3D models that resemble the furniture more.
- Implement scaling of the objects in Unity according to the bounding box.
- Implement orientation of the objects in Unity according to the bounding box.
- Configure Hydra to generate more accurate bounding boxes.
- Generate self-recorded data as input to Hydra.
- Implement correct labeling of objects.

5.2 Develop

The first part of the development phase consisted of working on setting up the camera to generate self-recorded input to Hydra. A GitHub issue in the Hydra repository regarding running Hydra on custom data [11] was read. Reading the issue, it was stated that the code for setting up the 2D semantic segmentation network for use with Hydra had not been publicly released. Consequently, generating custom input with a camera was not implemented successfully. However, in collaboration with a team at Ericsson, the Aria Digital Twin dataset was explored in order to determine if it could be converted to input to Hydra. Converting the files and generating a ROS bag file, it was successfully ran with Hydra. However, the position of the camera scanning the environment was not successfully implemented, and solely a small part of the environment was processed by Hydra. A visualization of the Aria Digital Twin dataset running with Hydra can be seen in Figure 5.2.

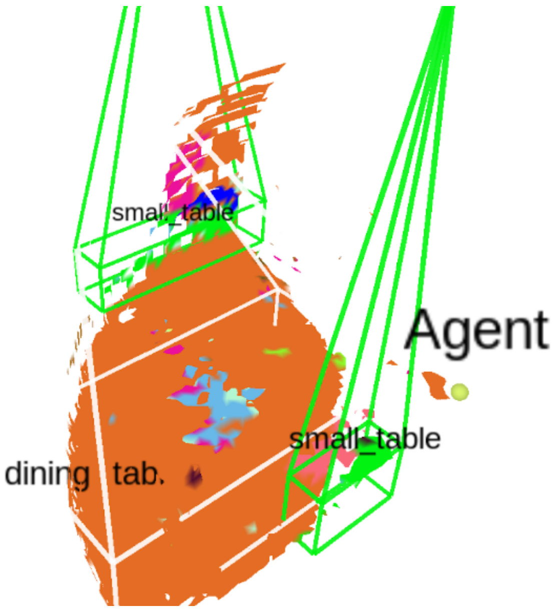


Figure 5.2: The Aria Digital Twin dataset running with Hydra.

Further, development regarding retrieving data about the orientation of objects was performed. By enabling oriented bounding boxes in Hydra, the bounding boxes were rotated according to the minimal bounding box possible, encapsulating its corresponding object. However, as the scaling, separation and merging of objects became less accurate using oriented bounding boxes, bounding boxes without orientation was used. Accordingly, no information about the orientation of the objects encapsulated by the bounding boxes was retrieved.

Following, a takeaway from the first prototype was that the labeling of the objects according to Hydra was incorrect but consistent. The problem was solved after posting an issue to the GitHub repository in which Hydra was published. An answer regarding the incorrect labeling was given and an update was done to the repository by one of the authors [12].

The basic white blocks in Unity representing the objects from Hydra from the first iteration were replaced with more realistic 3D models that resembled the type of object according to the label of the bounding box from Hydra. The 3D models were scaled according to the bounding box data from Hydra. Accordingly, the 3D models in Unity were scaled in real-time as Hydra ran.

Hydra was then configured in order to improve the bounding boxes in Hydra. As Hydra consisted of many files and lacked documentation, the task of configuring Hydra was not trivial. The code base was explored in order to identify the parts that potentially could affect the configuration of the bounding boxes. Different values of variables affecting the bounding boxes were arbitrarily tested, and the output of Hydra was visually analyzed in order to find improvements. A configuration appropriately balancing the separation and merging of objects was found. An image of the output in Hydra can be seen in Figure 5.3.

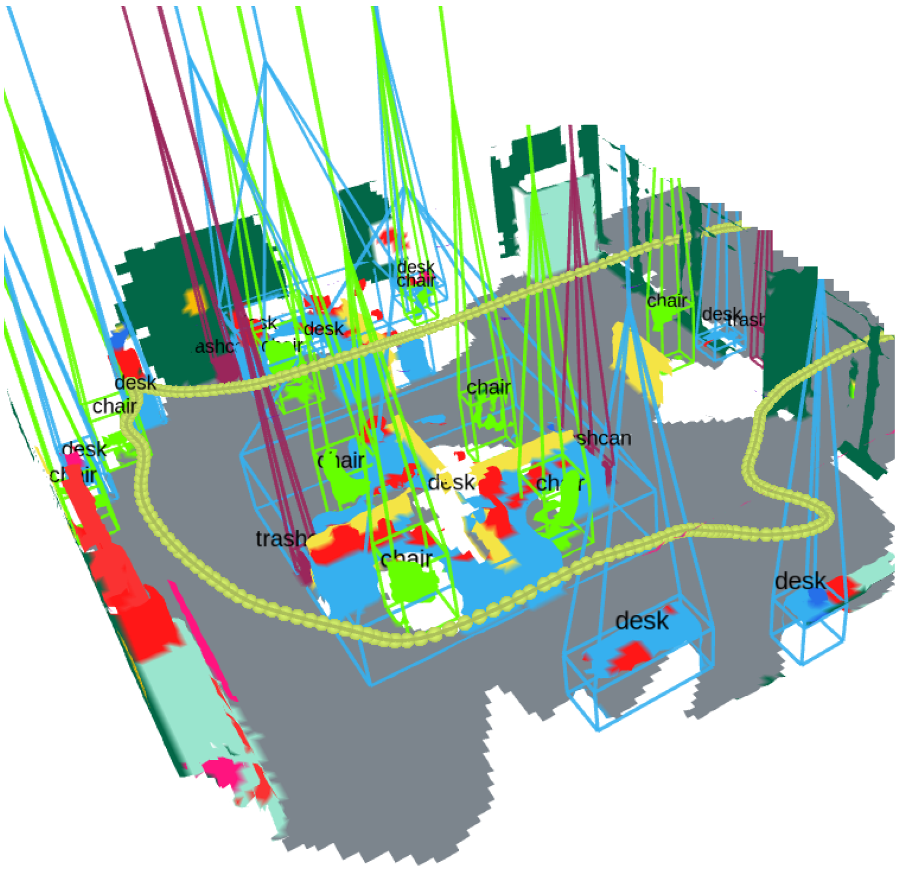


Figure 5.3: The office scene from the uHumans2 dataset after configuring Hydra.

The outcome of the second iteration was a spheroid with human-sized proportions moving according to the positional data from Hydra, leaving a purple trails of its trajectory. Objects were generated in Unity according to the specifications of the bounding boxes from Hydra, containing information about the objects position, scale and label. 3D model prefabs were used in order have a more clear resemblance of the furniture. The 3D models were also colorized in order to make them visually stand out more from each other.

An overview of the implemented parts can be seen in Figure 5.4. The Hydra node was the ROS node started when running Hydra. The Hydra TCP Bridge consisted of two nodes: the ROS node subscribing to the positional data from the camera in Hydra and the ROS node subscribing to the ROS topic in Hydra where bounding box data was sent. The ROS TCP Connector was the ROS node enabling communication with Unity. The App was the ROS node that subscribed to the data transmitted over the ROS TCP Connector, in order to get information about the camera position and the bounding boxes of the objects from Hydra. As input to Hydra, the office scene from the uHumans2 data set was used. An image from the Unity application can be seen in Figure 5.5.

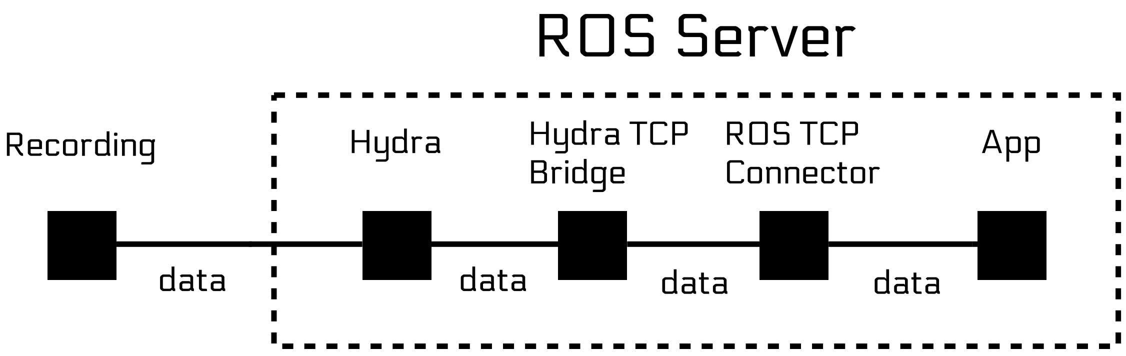


Figure 5.4: Overview of the implemented workflow.

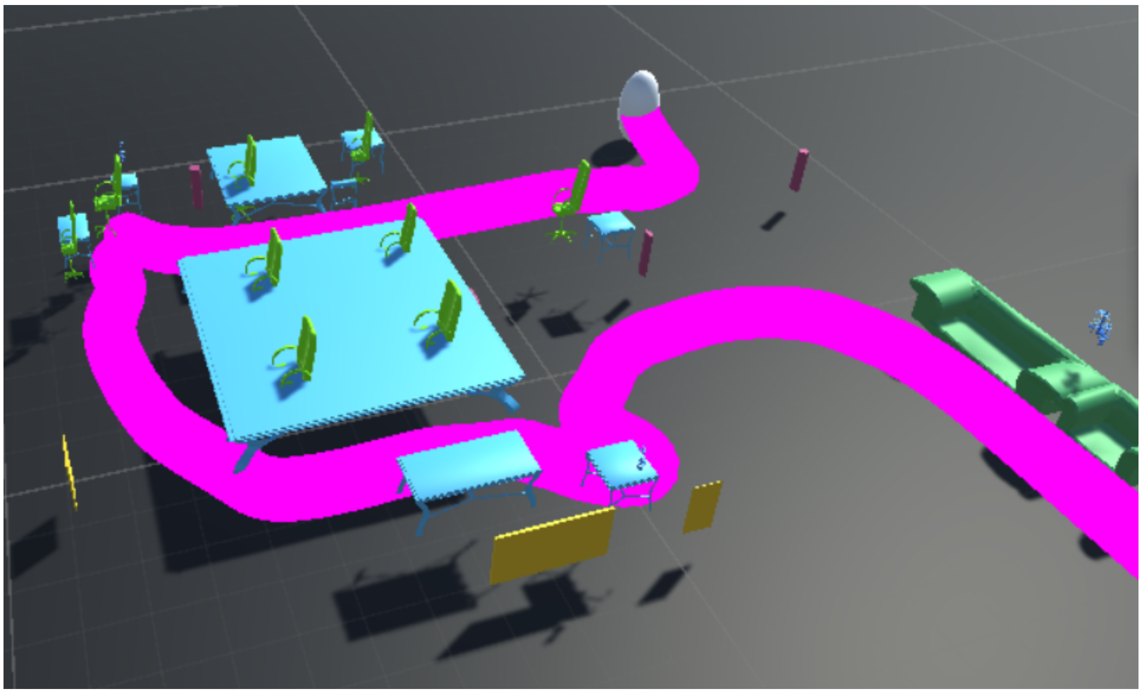


Figure 5.5: Unity application after the second iteration. The sphere illustrates the agent, with its trajectory visualized by the purple trail. The colored 3D models illustrates the detected objects.

5.3 Takeaways

After developing the second iteration of the workflow some takeaways to consider for the following iteration of the development was written. The takeaways were:

- If enabled, Hydra provides oriented bounding boxes in order to minimize the volume of the bounding box while still encapsulating the entire object. However, the scaling, separation and merging of objects became less accurate using oriented bounding boxes. Calculating the orientation of the actual object may not be prioritized for the following iteration.

- The scaling of the objects from Hydra to Unity works well.
- The configuration of Hydra was a trade-off between striving for fully encapsulating the objects, without merging too many objects.
- Hydra can be configured in many ways, and it may be explored further.
- Getting the camera to work to generate self-recorded input to Hydra is difficult and may not be prioritized for the following iteration.
- The successful running of Hydra with Project Aria Digital Twin dataset may be a step in the right direction to run Hydra with self-recorded data from a camera.

Chapter 6

Iteration 3

This chapter describes in detail the third iteration of the project process, including the stages involved and the specific activities for each stage. The designing and the development of the third prototype was based on the double diamond model, with the modification that the discover phase was omitted, as seen in Figure 6.1. Lastly, some final takeaways of the iteration is presented.

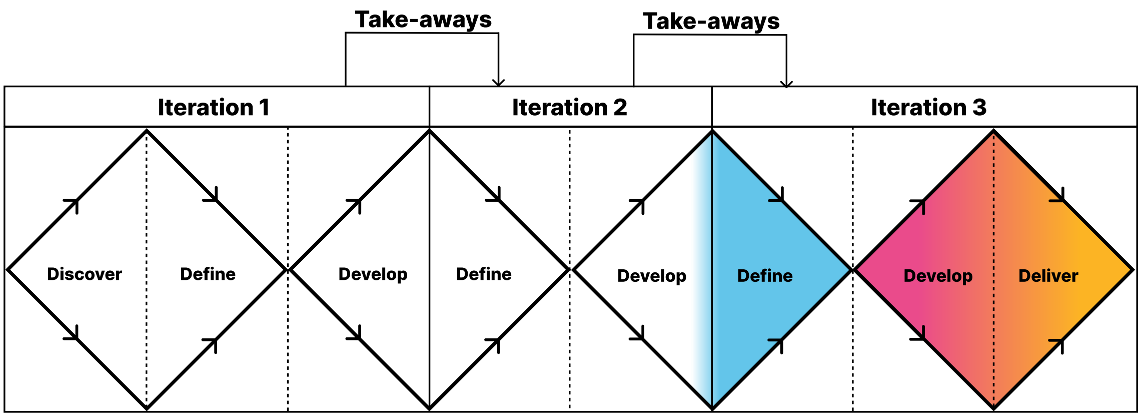


Figure 6.1: The stages of development involved in iteration 3.

6.1 Define

The second prototype was used as a basis when designing and developing the third prototype. A test plan had also been written prior to developing the third prototype, it was used when defining the third prototype in order to make it fit well with the user tests. The test plan is described in Chapter 7.

The concept to strive for when developing the third iteration of the prototype was defined as:

- More realistic 3D models.
- Improve the height of the objects.
- Generate floor, walls and ceiling in the environment.
- Implement an XR integration in Unity.
- Pre-record a ROS bag file to run during the real-time rendering of the test.

6.2 Develop

The development phase began with recording a ROS bag file to run during one part of the test where the Unity environment was to be rendered in real-time. The recording started from the beginning of the uHumans2 office scene and was stopped when the camera reached the end of the first room of the scene.

The textures of the 3D models in Unity was updated to be more realistic. The 3D models used was a chair, a desk, a trashcan and a ceiling lamp. The human-sized spheroid that represented the camera in the Unity scene was replaced with a 3D model resembling a camera.

Two different Unity scenes were created. For the first scene, the ROS bag file was ran and the 3D models according to the bounding boxes in Hydra was generated in Unity. The scene then consisted of a room in an office with chairs, trashcans, desks and ceiling lamps. For the second scene the same room was used. However, the 3D models were deleted in order to let Hydra run with the pre-recorded ROS bag file, sending the data to Unity through the data pipeline, generating the objects in Unity according to the bounding boxes in real-time.

For both scenes, a floor, ceiling and walls was manually implemented in Unity, without using Hydra. This was done because of the inability to find a configuration of Hydra that managed to successfully generate well-defined bounding boxes for furniture together with the room definitions. Accordingly, the configuration of Hydra was set with the ambition to improve the bounding boxes of the furniture, not the room definitions. As a result, the room definitions was decided to be manually implemented.

Lastly, an XR integration was set up in Unity for both scenes in order to let the user experience the scenes in XR.

6.3 Deliver

The outcome of the third iteration was two different Unity scenes to run during the user tests. One scene to test the users' spatial awareness and one scene to test the users' experience of being present in XR in a virtual environment being rendered in real-time. The first scene consisted of an office room with chairs, desks, trashcans and ceiling lamps. The second scene consisted of the same room definitions, but was empty of furniture. A ROS bag file was recorded that when ran with Hydra, sending the bounding box data to Unity through the data pipeline, spawned a camera in the Unity scene that moved around, rendering objects in real-time. The camera followed the trajectory of the camera from the pre-recorded ROS bag file. In both scenes, one could move around freely using the hand controllers of the HMD. Either by teleportation using the joystick of the right hand controller, or by locomotion using the joystick of the left hand controller. The delivered concepts included:

- 3D model prefabs with realistic textures.
- 3D model camera prefab representing the camera scanning the environment.
- Fixed height of the objects in Unity.
- Added room definition in Unity (floor, walls, ceiling) manually.
- XR integration in Unity.
- Pre-recorded ROS bag file to run during the real-time rendering during the user tests.

An overview of the implemented parts can be seen in Figure 6.2. An image from the first scene of the Unity application can be seen in Figure 6.3. Images from the second scene of the Unity application can be seen in Figure 6.4, Figure 6.5, Figure 6.6 and Figure 6.7.

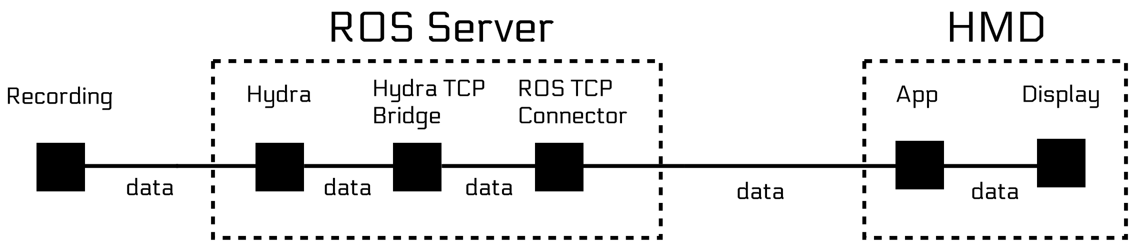


Figure 6.2: Overview of the implemented workflow.

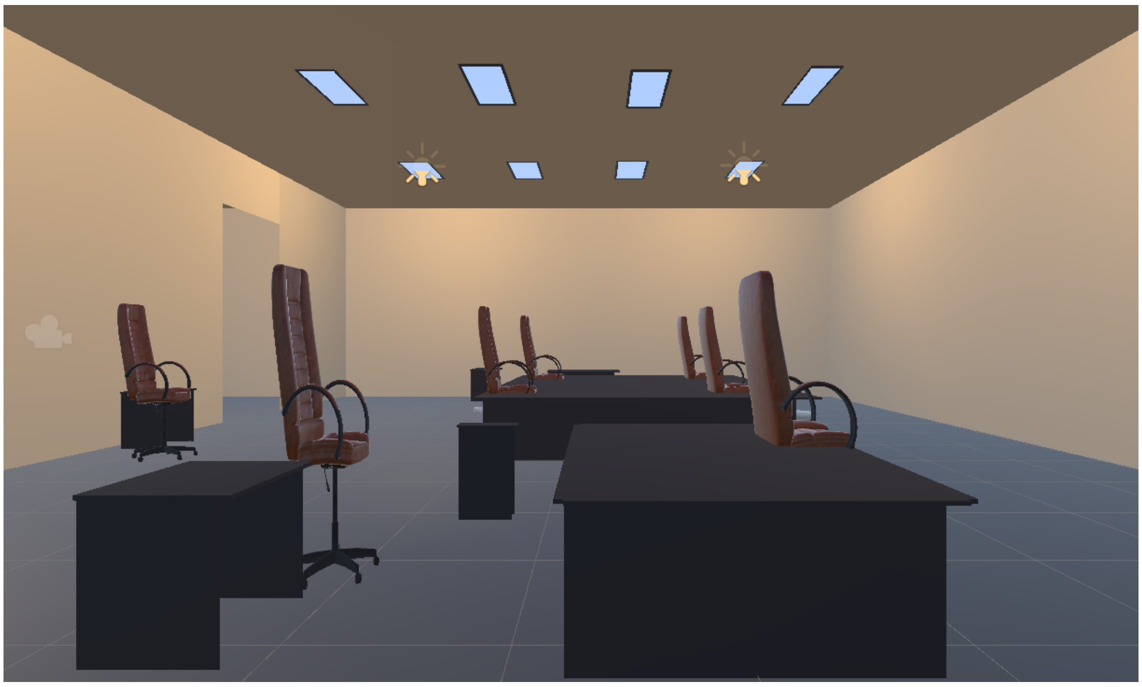


Figure 6.3: The first scene of the Unity application after the third iteration. The objects were rendered when the scene was initialized.

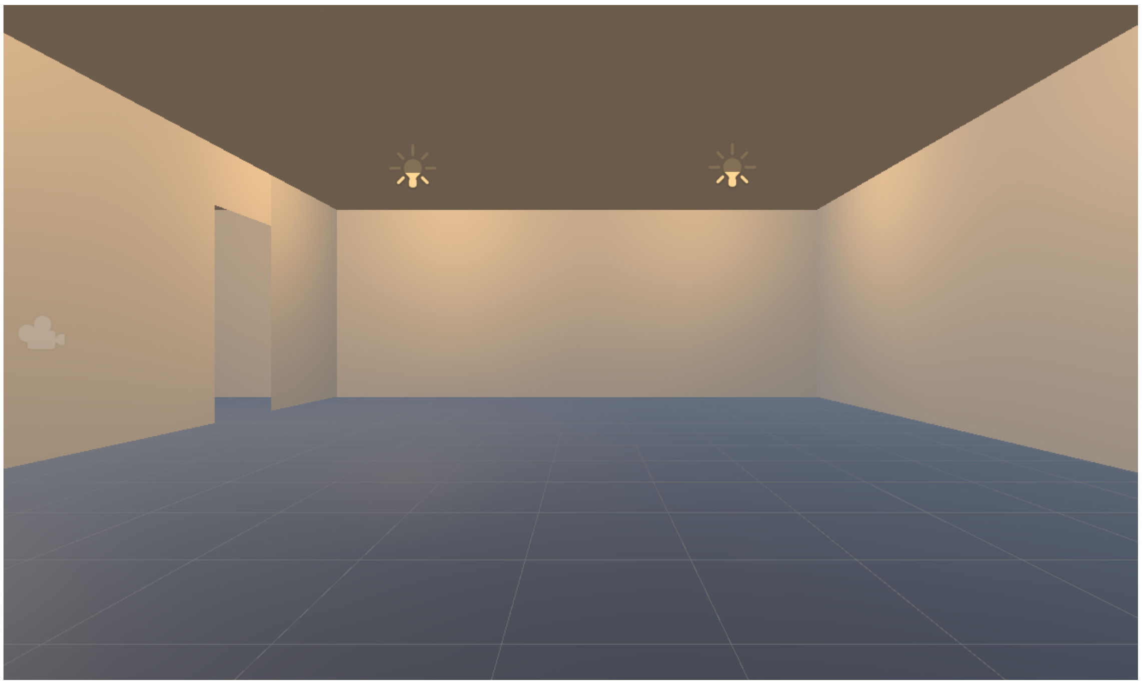


Figure 6.4: The second scene of the Unity application after the third iteration. Before the rendering of the objects, the scene was empty of furniture.

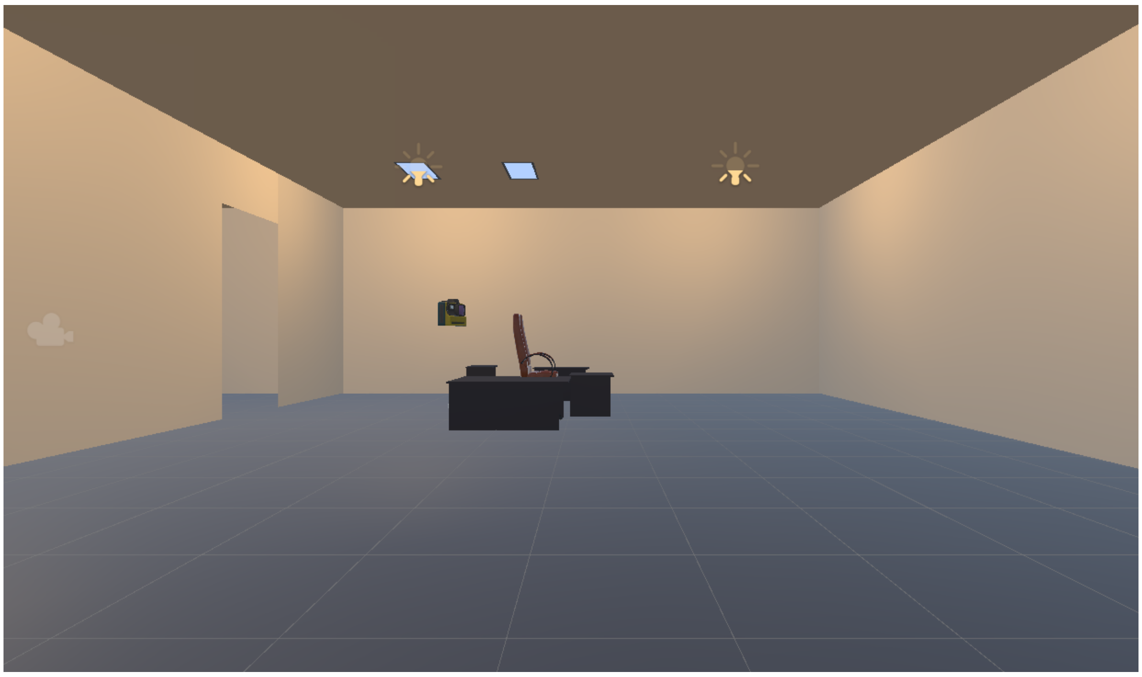


Figure 6.5: The second scene of the Unity application after the third iteration. As Hydra ran with the pre-recorded ROS bag file, a camera moved around the scene, rendering objects in real-time.

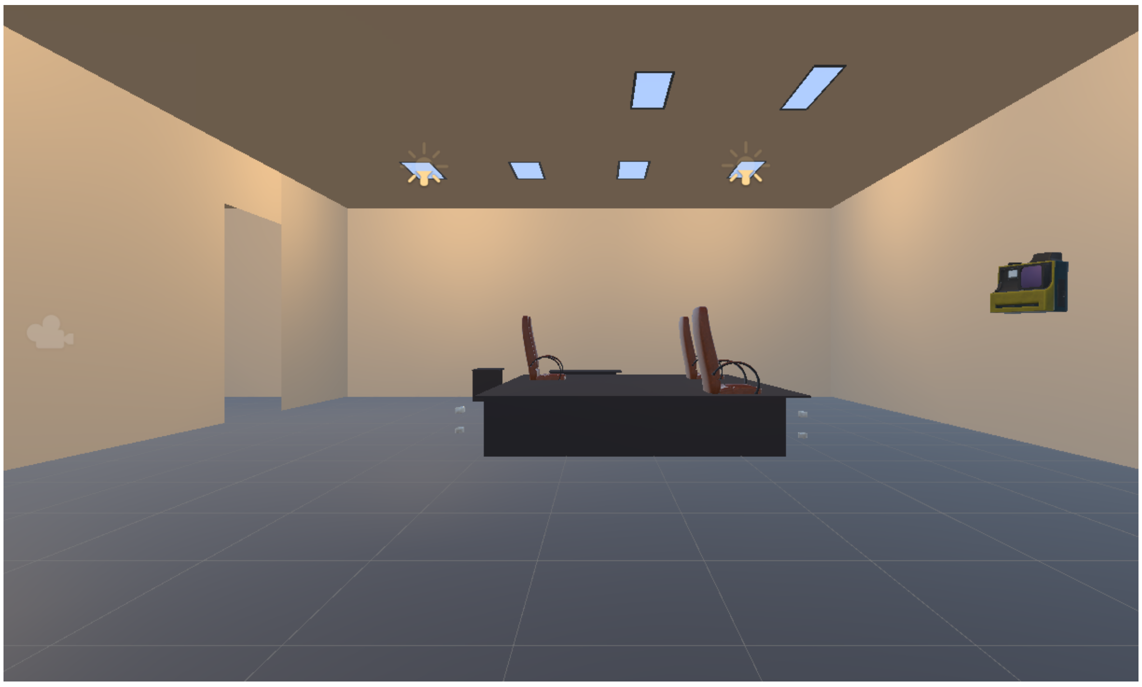


Figure 6.6: The second scene of the Unity application after the third iteration. The camera moved around the room, rendering objects in real-time.

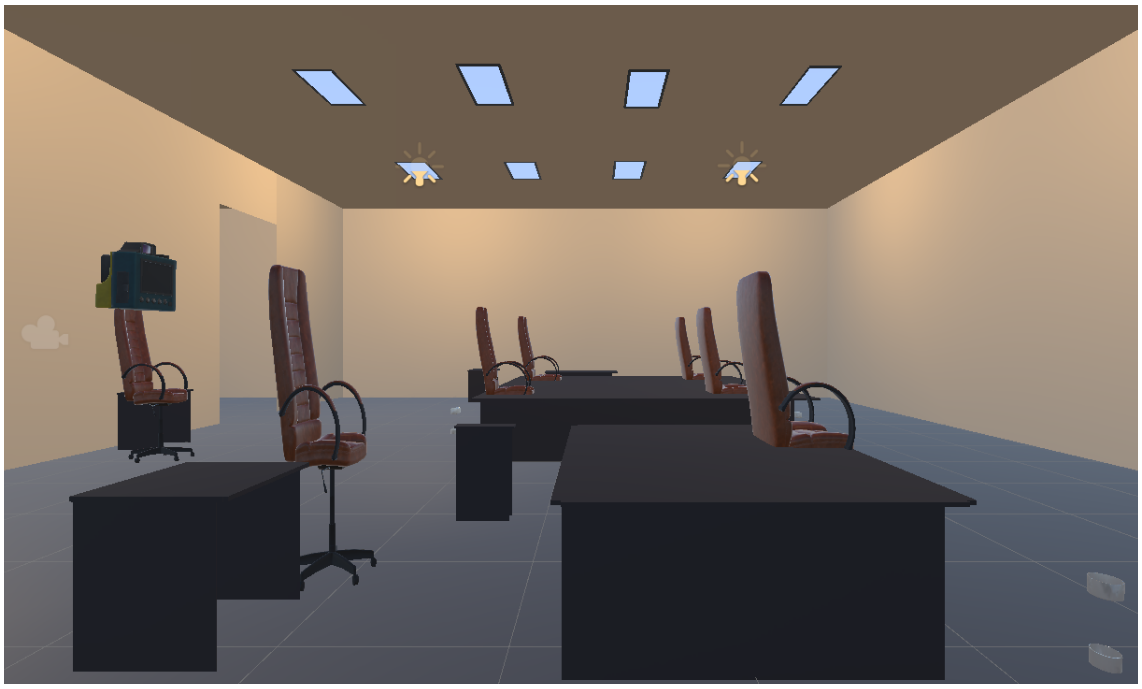


Figure 6.7: The second scene of the Unity application after the third iteration. When the pre-recorded ROS bag file ended, the camera stopped and it was done rendering the objects.

Chapter 7
User Testing

This chapter describes in detail the user testing of the project process. The participants and their attributes are described, the test setup is defined and the test procedure is described. Lastly, the results are presented.

7.1 Test Plan

The purpose of performing the user tests was primarily to answer **RQ1**: *How is a user's spatial awareness affected in a virtual environment reconstructed from a real environment with Hydra?* and **RQ3**: *How does the user perceive the virtual environment while objects are rendered in real time?*. The first scene from the prototype corresponded to **RQ1**, as the furniture was rendered before the user entered the XR environment. The second scene from the prototype corresponded to **RQ3**, as the furniture was rendered in real-time as the user was present in the XR environment.

7.1.1 Selection of Participants

The criteria for the selection of participants:

- The participants had to have some experience with XR, i.e. tried it before.
- There should be roughly the same number of participants in the two test groups.

The first criterion was set to facilitate participants who would be more focused on the actual test and having experiences to compare it to. The second criterion was chosen so a comparison between participants who worked in related areas or with technologies similar to the implemented one, and to broaden the perceptive of the results. Furthermore, the two groups represented stakeholders, i.e. the first group, Ericsson employees represented product owners, managers or partners and the other group represented potential future users. The two groups were also studied as a whole.

In total the study involved 11 participants, 5 from Ericsson and 6 with various backgrounds, see Figure 7.1 and the age and gender distributions in Figure 7.2. To verify that the first criterion was met, potential participants had to fill out a form where they, among other things, were asked to rate their level of previous experience with XR. The self-assessed experience was on a scale from zero to five, were zero meant no experience and five meant considerable, see Figure 7.3.

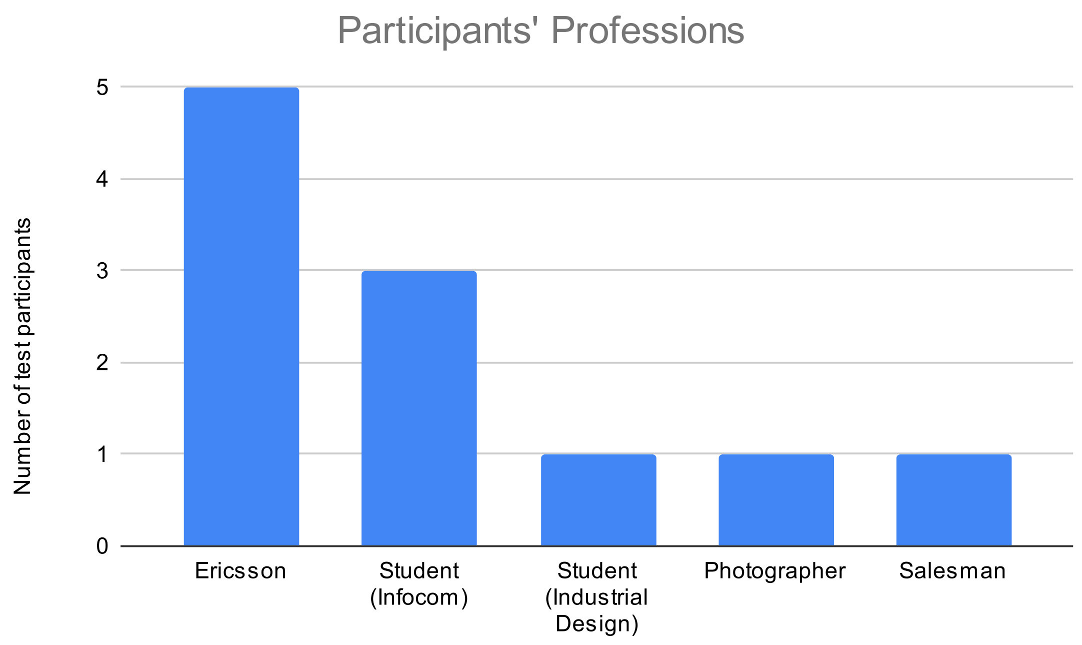


Figure 7.1: Distribution of participants' professions.

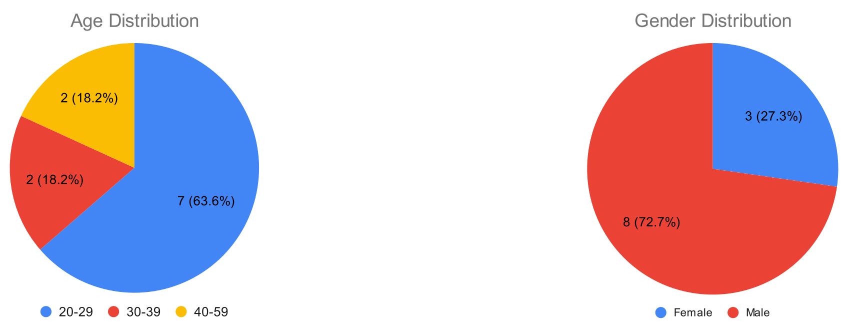


Figure 7.2: Age distribution (left) and gender distribution (right).

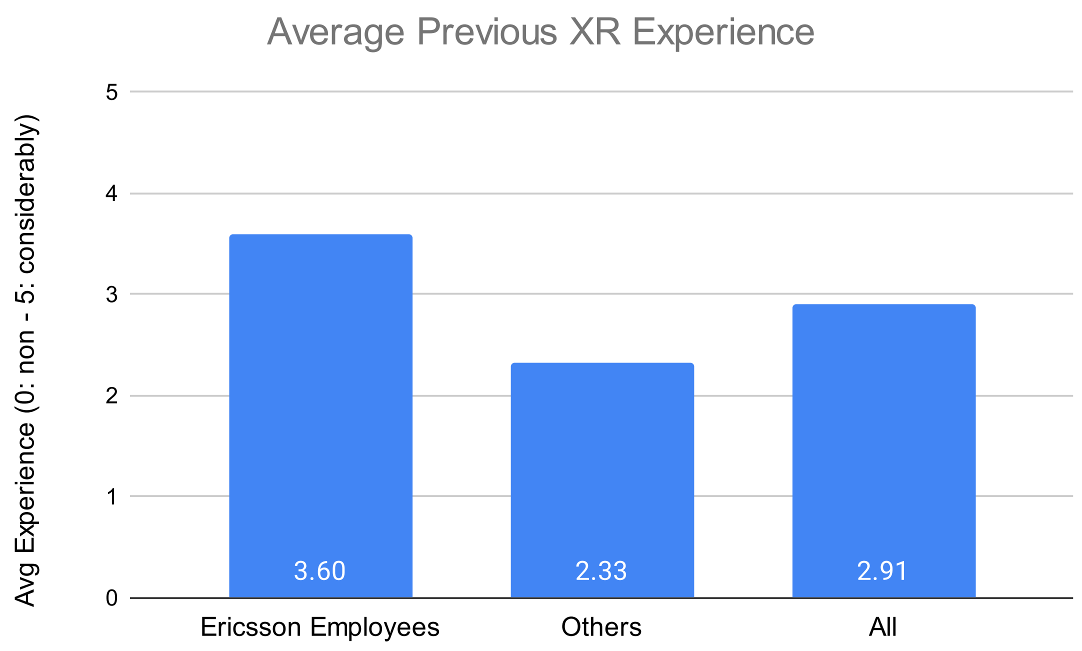


Figure 7.3: Self-assessed average of previous XR experience on a group level.

7.1.2 Roles

Two roles were needed during the test session, the test moderator and the data gatherer.

Test Moderator

The test moderator had the main responsibility during the test session. Before each test session the moderator needed to verify that the XR headset and demo were working, to prevent disruptions occurring during the test. In addition, the responsibility was to check that other material for the test was available, this included informed consent, orientation script (see Appendix B), floor plans and debriefing document (see Appendix D).

During the test session, only the moderator would interact with the test person. This meant confirming that the participant understood, read and signed the informed consent, reading the orientation script, answering potential questions and finally carrying out the debriefing.

Data Gatherer

Throughout the testing session, the data gatherer was tasked with overseeing all aspects of data collection. The main responsibilities entailed audio recording, general observation notes, time tracking, logging participants performance and writing down answers from the debriefing interview. The tools used by the data gatherer were a phone for audio recording, a stopwatch and an observation protocol (see Appendix C). The purpose of the observation protocol was to log all information gathered during the test, e.g. participant task information such as time, guesses and whether the guesses were correct or not. The audio recordings functioned as a fail-safe in the case of missed information or if it was an uncertainty regarding a participant's answer or comment.

7.1.3 Test Setup

Test Environment

Two different locations were used for the test sessions. The first one was a conference room at Ericsson's office building in Lund and the second one the UX lab at IKDC. In order to maintain a consistent test setup across both locations, the full capabilities of the UX lab, i.e. the control room behind a one-way mirror equipped with cameras and microphones, were not utilized.

Test Equipment

The test equipment used by the participants were three A4-papers with the floor plans (see Appendix E) and the Meta Quest 2 with two hand controllers.

The test moderator used a laptop to run the XR-demo, a Quest Link cable, an orientation script and a debriefing document.

The data gatherer used a laptop for the observation protocol, a stopwatch, and a phone to record audio. The final test setup is illustrated in Figure 7.4.

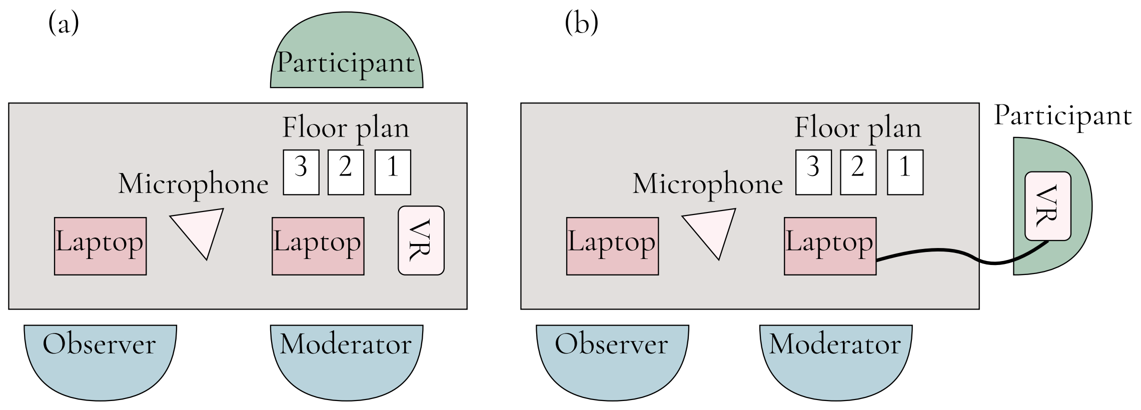


Figure 7.4: Illustration of test setup before XR demo (a) and during the XR demo (b).

7.1.4 Procedure

With the goal of ensuring consistency in all tests, the same procedure was repeated for each test.

The test person arrived and was greeted by the test moderator and gatherer. The participant was then asked to sit down in the chair in front of the moderator. The moderator handed the informed consent to the participant, and explained that the participant should read it through, check the boxes that the participant agree to and finally sign it. If the participant agreed then the data gatherer started recording the session, while the moderator started reading the orientation script.

The moderator placed the three floor plans (see Appendix E) in front of the participant and told the participant to try to memorize them. The participant was then asked to stand up and was instructed on how to put on the HMD and how to move around with the controllers (either with teleportation or joystick locomotion). When the HMD was comfortably in place, the moderator put on the first XR demo (see Figure 6.4-6.7. The participant was not given instructions on how many times they could guess which floor plan the environment represented and not told if a guess was correct. They were given at most five minutes.

The second demo was then started. The only task the participant had was to explore the environment and try to think out loud. The participant could stop whenever they wanted, but at most 5 minutes were given.

The participant was then asked to sit down again. The test session was finished with the debriefing interview.

7.1.5 Pilot Test

To verify that the test plan worked as intended and to find potential issues with it, a pilot test was carried out. A participant was selected and underwent the entire test procedure. The test yielded some areas that needed to be addressed. **Takeaway 1**: the cable between the laptop and the HMD needed to be further reinforced to avoid interruptions during the tests. This was addressed by adding another hook-and-loop fastener (Velcro cable tie) between the cable and the HMD. **Takeaway 2**: the laptop had limited performance capabilities which became a bottleneck when running the whole hydra workflow. This was addressed by pre-recording a ROS bag file with the real-time Hydra topics which were then played with the remaining workflow.

7.1.6 Data Collection

The data collection served an important purpose in order to answer the research questions outlined in Section 1.3.

Participation Form

Two participation forms were used to handle the booking of participants, one for the Ericsson employees and one for the other group. In addition to handling the bookings it also included questions to gather information of the persons age, gender and self-assessed XR experience. This could then be used to ensure that the criteria, outlined in Section 7.1.1, were met.

Informed Consent

Before any data was collected during the test, participants needed to be informed and give consent to what and how their data would be handled. This was done with an informed consent form that needed to be signed by the moderator and participant before any collection began.

Audio Recordings

If the participant consented to being audio recorded the observer started a voice memo recording on their phone and pointed the microphone towards the participant. As mentioned in Section 7.1.2, the recording played an important part in ensuring that no data was lost without unnecessary interruptions.

Observation Protocol

The observation protocol was used to collect both qualitative and quantitative data. The qualitative data included information about how the user experienced the environment, reconstruction and scanner. The information was categorized by predefined parts of the test, i.e. during the orientation, first part of the demo and the second part of the demo. The quantitative data was only gathered from the first part of the test, when the participant tried to understand which floor plan the environment referred to. The information logged was which floor plan was guessed, timestamp, correct or wrong and number of times the participant looked at the floor plans, i.e. took off the HMD to see the floor plans.

Debriefing

The test session was concluded with a debriefing in the form of a semi-structured interview. The questions can be found in Appendix D. The questions consisted of two evaluation scales as well as multiple open ended questions. Despite resulting in different types of answers, i.e. quantitative and qualitative, they still shared the commonality in their purpose of providing subjective data on various aspects of the test.

The first evaluation scale was to rate how well the virtual environment resembled the floor plan on a scale from 1 to 5 (1: not at all, 5: identical). The second one to assess the difficulty of understanding the correct floor plan on a scale from 1 to 5 (1: very difficult, 5: very easy).

7.2 Results

The results from the tests are divided in two parts, quantitative and qualitative results. The quantitative results include numerical data collected from test users performance, e.g. success rate, time to guess, as well as rating different aspects of the of test experience from 1-5. The qualitative results consists of data gathered both during the test, i.e. test users comments, and from the debriefing interview in the end. The data was then used to create an affinity diagram, to get a deeper understanding and to identify potential patterns from the thoughts and answers.

7.2.1 Quantitative Results

Participants

The background information of the participants was collected from the participation form described in Section 7.1.6. In total, eleven people participated in the test, five employees from Ericsson and six from the other group. Out of the Ericsson group three assessed them as having considerably (high rating between 4-5) XR experience, one as moderate experience (mid rating of 3) and the last one as having limited experience (low rating between 1-2). In the other group only one out of the six participants rated them self as having considerable XR experience. In this group only one had a mid rating and the majority of the group considered them self to have limited XR experience. A chart of the experience distribution can be found in Figure 7.5.

The ages of the participants were also gathered from the form. The Ericsson group consisted of participants between 27 and 59 years old, with an average age of ~40 years (= 39.8). While the other group were between 25 and 28 years old and an average age of ~27 years (= 26.5). See Table 7.1.

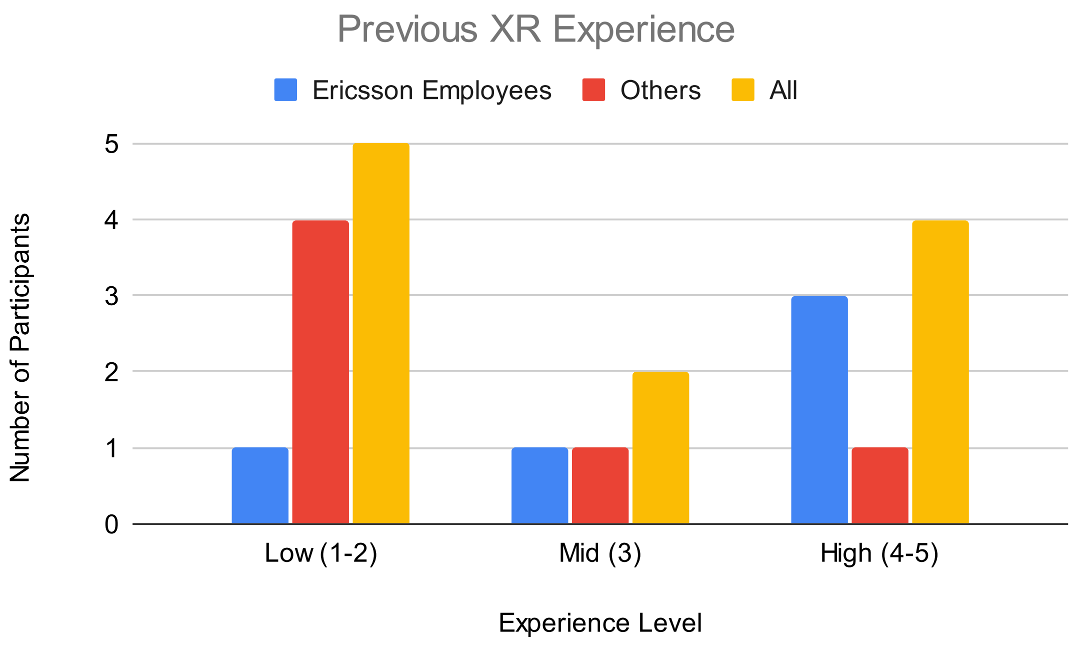


Figure 7.5: Self evaluated average of previous XR experience on a group level.

Table 7.1: Table with participant ages and group average.

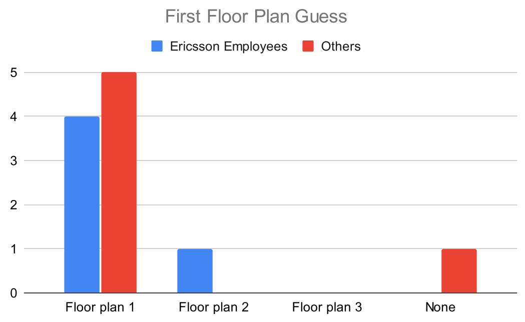
Ericsson Employees	Age
T1	27
T2	36
T3	32
T4	45
T5	59
Average	39.8

Others	Age
T6	25
T7	28
T8	25
T10	26
T11	27
T13	28
Average	26.5

Success Rate and Time Duration

The results for success rate and time duration were collected as a part of the observation protocol. Every time a participant made a guess of which floor plan depicting the room that the 3D environment had been reconstructed from, the timestamp, floor plan guess and if it was correct or not it were logged in the observation protocol. Furthermore the number of times the participants looked at the floor plan during the test was also added. The participants looked at the floor plan at the most 2 times. From the Ericsson group three of the five participants looked at the floor plans again, and in the other group none of the six participants looked again.

In Figure 7.6 the left one corresponds to the results of the first guess. Floor plan 1 was the correct floor plan and as can be seen in the figure, four out of the five Ericsson participants guessed correctly and for the other group five out of the six participants were also correct. The right chart was the results from the second guess. An important note for this graph, guesses made by participants that only made one guess were still counted as the second guess to better illustrate changed made in the two groups. Only one participant in the Ericsson group changed their answer, i.e. by the second guess all participants from Ericsson had guessed the correct floor plan. The participants from the other group guessed on the same floor plan they had in the first guess, which can be seen in the right chart in Figure 7.6. The average success rate for the first and second guess were then calculated as correct answers divided by the number of participants. The result shows a small increase from the first to second guess, which can be seen in Figure 7.7



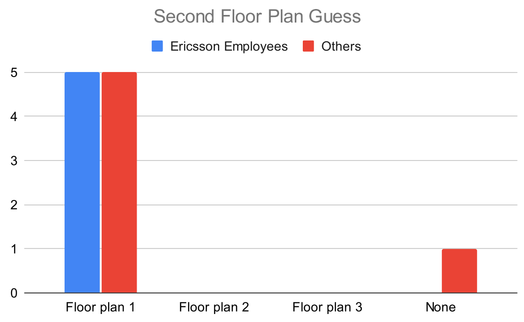


Figure 7.6: Distribution of guessed floor plan in each group.

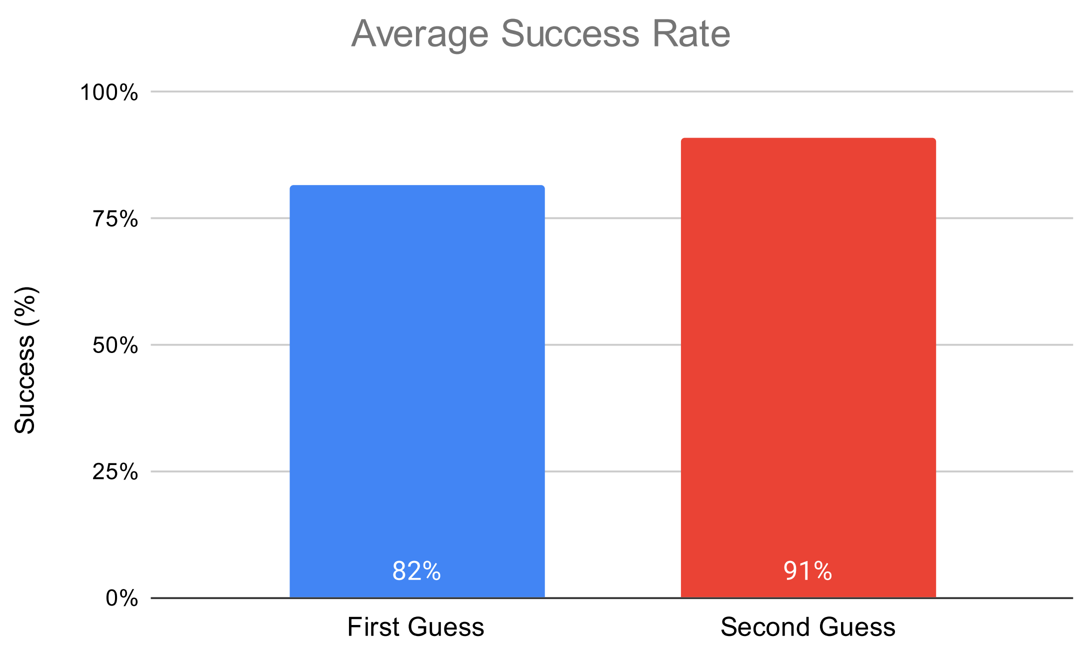


Figure 7.7: Average success rate in the groups and total.

The average time to guess was also calculated, see Figure 7.8. All participants made at least one guess, which was used to calculate the average time for the first guess. However only six participants made a second guess, three from each group, they either changed their answer or reaffirmed their previous one. The average for the second guess was calculated from these six timestamps. All data can be found in Table 7.2.

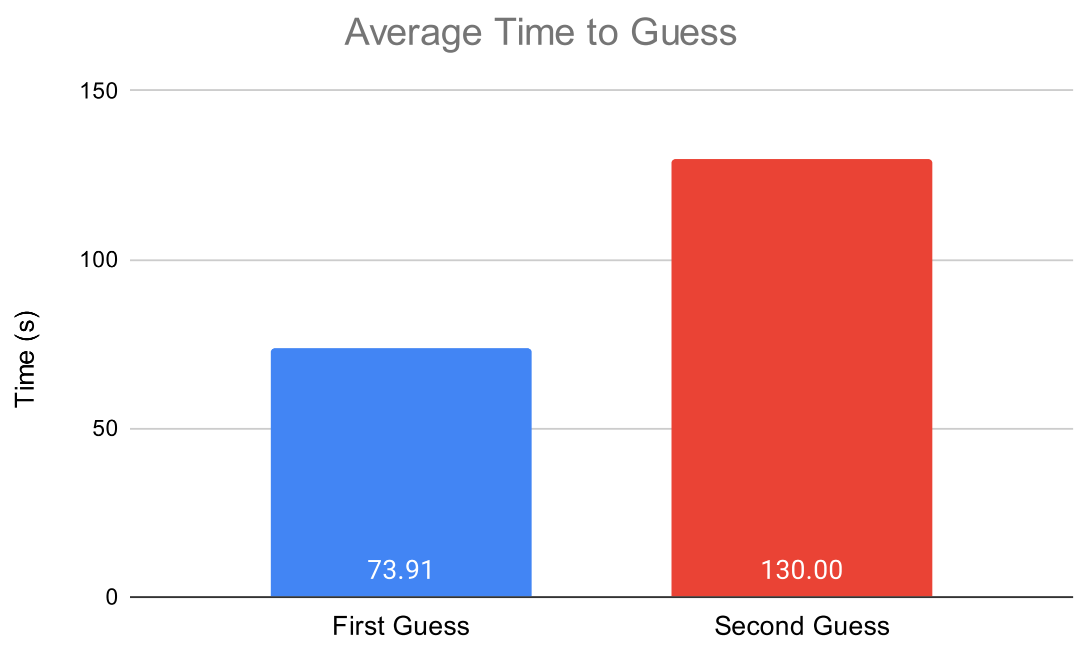


Figure 7.8: Average time in the groups and total.

Table 7.2: FP = floor plan. Table with data from both guesses, first guess (left) second guess (right). "-" represents no guess.

Participant	Time (s)	FP nbr	Correct	Participant	Time (s)	FP nbr	Correct
T1	190	1	Yes	T1	260	1	Yes
T2	180	2	No	T2	181	1	Yes
T3	110	1	Yes	T3	150	1	Yes
T4	23	1	Yes	T4	-	-	-
T5	14	1	Yes	T5	-	-	-
T6	40	1	Yes	T6	70	1	Yes
T7	167	0	No	T7	-	-	-
T8	5	1	Yes	T8	60	1	Yes
T10	26	1	Yes	T10	59	1	Yes
T11	20	1	Yes	T11	-	-	-
T13	38	1	Yes	T13	-	-	-

Resemblance and Difficulty Evaluation

The results from the the first part of the debriefing, were the participants rated how well the virtual environment resembled the correct floor plan and how difficult they thought it was to identify the correct floor plan, were divided into two each. The first as an average for each group and the second based on experience (low, mid or high. In Figure 7.9, the average experienced resemblance and Figure 7.10, the average experienced difficulty.

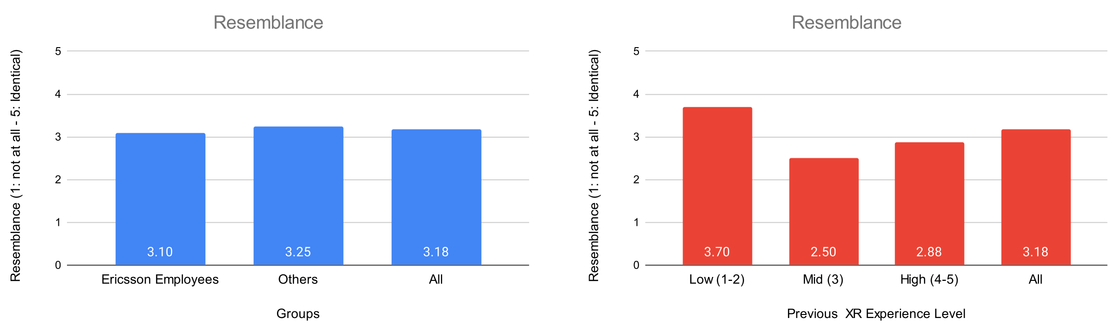


Figure 7.9: The participants' ratings of resemblance between floor plan and environment.

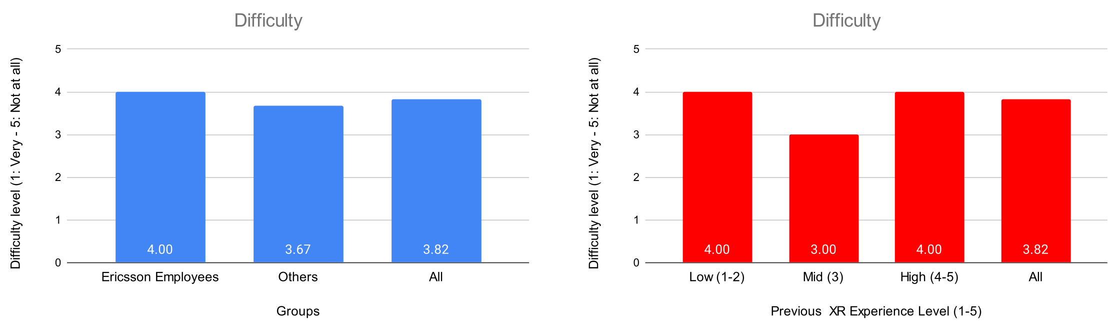


Figure 7.10: Self evaluation of difficulty in task to identify correct floor plan corresponding to the environment.

7.2.2 Qualitative Results

The data gathered during the tests, i.e. comments during the test and answers from the debriefing, was transferred to individual notes in Figma. Minimal changes were made to some notes to fit the format better. From the large amount of unsorted data an affinity diagram was created by using the method described in Section 3.2.2. The process was performed three times, first general sorting into categories, then subcategories was added and refinements was done, and lastly reducing it down by restructuring and removing duplicate statements. The result can be found in Figure 7.11, 7.12 and 7.13. An addition to the affinity diagrams was that some participants either asked or tried to point out where in the floor plan they had started or where they stood when finishing the second test case. One participant pointed to the correct position.

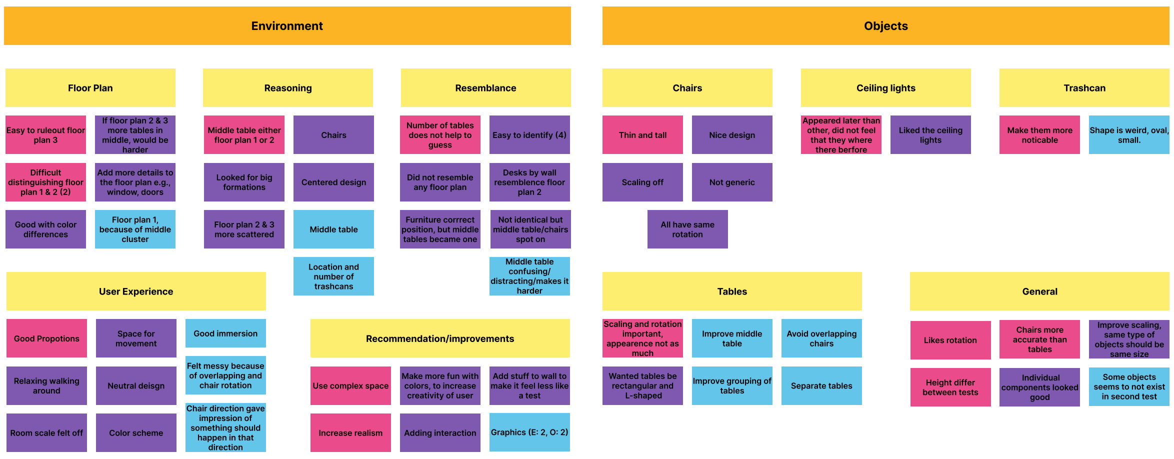

Environment
Floor Plan
Easy to ruleout floor plan 3
If floor plan 2 & 3 more tables in middle, would be harder
Difficult distinguishing floor plan 1 & 2 (2)
Add more details to the floor plan e.g., window, doors
Good with color differences
Floor plan 1, because of middle cluster
Reasoning
Middle table either floor plan 1 or 2
Chairs
Looked for big formations
Centered design
Floor plan 2 & 3 more scattered
Middle table
Location and number of trashcans
Resemblance
Number of tables does not help to guess
Easy to identify (4)
Did not resemble any floor plan
Desks by wall resemblence floor plan 2
Furniture corrrect position, but middle tables became one
Not identical but middle table/chairs spot on
Middle table confusing/ distracting/makes it harder
User Experience
Good Propotions
Space for movement
Good immersion
Relaxing walking around
Neutral deisgn
Felt messy because of overlapping and chair rotation
Room scale felt off
Color scheme
Chair direction gave impression of something should happen in that direction
Recommendation/improvements
Use complex space
Make more fun with colors, to increase creativity of user
Add stuff to wall to make it feel less like a test
Increase realism
Adding interaction
Graphics (E: 2, O: 2)
Objects
Chairs
Thin and tall
Nice design
Scaling off
Not generic
All have same rotation
Ceiling lights
Appeared later than other, did not feel that they where there berfore
Liked the ceiling lights
Trashcan
Make them more noticable
Shape is weird, oval, small.
Tables
Scaling and rotation important, appearence not as much
Improve middle table
Avoid overlapping chairs
Wanted tables be rectangular and L-shaped
Improve grouping of tables
Separate tables
General
Likes rotation
Chairs more accurate than tables
Improve scaling, same type of objects should be same size
Height differ between tests
Individual components looked good
Some objects seems to not exist in second test

Figure 7.11: Affinity diagram from data related to the environment and objects from the test session.

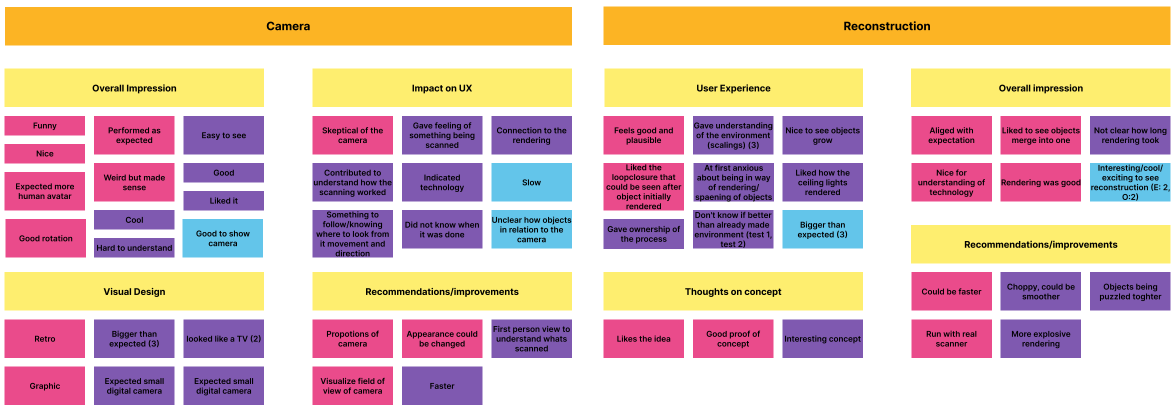

Camera
Overall Impression
Funny
Performed as expected
Easy to see
Nice
Expected more human avatar
Weird but made sense
Good
Liked it
Cool
Good rotation
Hard to understand
Good to show camera
Impact on UX
Skeptical of the camera
Gave feeling of something being scanned
Connection to the rendering
Contributed to understand how the scanning worked
Indicated technology
Slow
Something to follow/knowing where to look from it movement and direction
Did not know when it was done
Unclear how objects in relation to the camera
Visual Design
Retro
Bigger than expected (3)
looked like a TV (2)
Graphic
Expected small digital camera
Expected small digital camera
Recommendations/improvements
Propotions of camera
Appearance could be changed
First person view to understand whats scanned
Visualize field of view of camera
Faster
Reconstruction
User Experience
Feels good and plausible
Gave understanding of the environment (scalings) (3)
Nice to see objects grow
Liked the loopclosure that could be seen after object initially rendered
At first anxious about being in way of rendering/ spaening of objects
Liked how the ceiling lights rendered
Gave ownership of the process
Don't know if better than already made environment (test 1, test 2)
Bigger than expected (3)
Thoughts on concept
Likes the idea
Good proof of concept
Interesting concept
Overall impression
Aliged with expectation
Liked to see objects merge into one
Not clear how long rendering took
Nice for understanding of technology
Rendering was good
Interesting/cool/ exciting to see reconstruction (E: 2, O:2)
Recommendations/improvements
Could be faster
Choppy, could be smoother
Objects being puzzled toghter
Run with real scanner
More explosive rendering

Figure 7.12: Affinity diagram from data related to the camera and reconstruction from the test session.

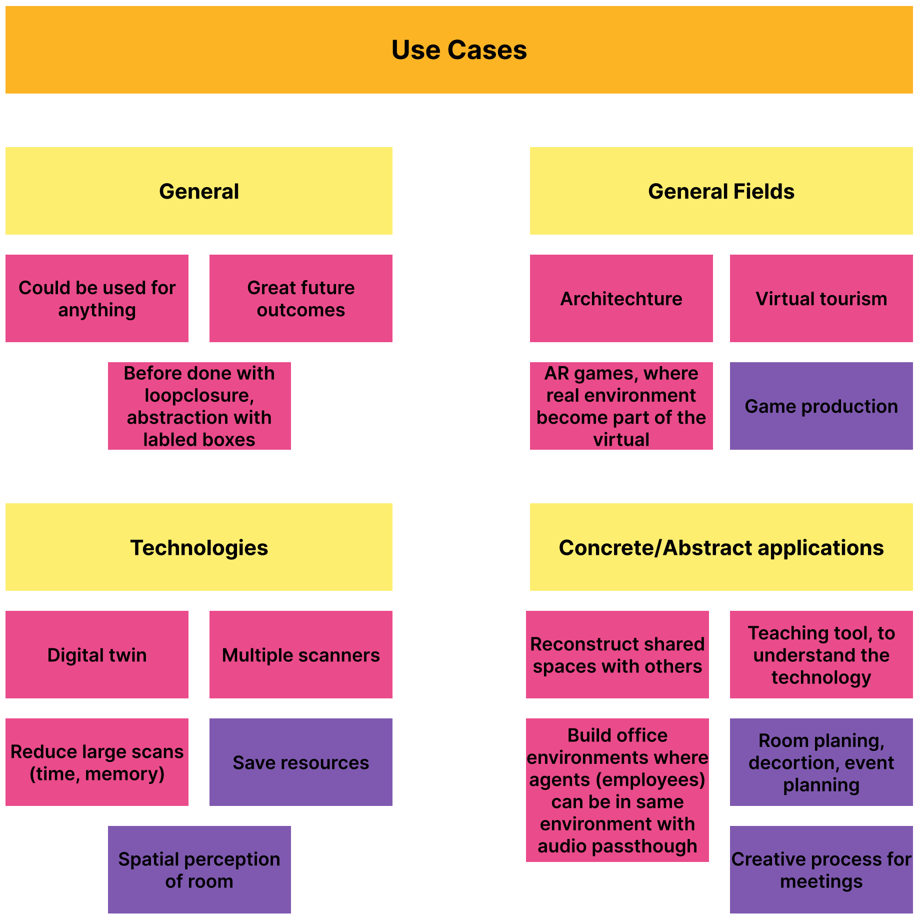

Use Cases
General
Could be used for anything
Great future outcomes
Before done with loopclosure, abstraction with labled boxes
General Fields
Architechture
Virtual tourism
AR games, where real environment become part of the virtual
Game production
Technologies
Digital twin
Multiple scanners
Reduce large scans (time, memory)
Save resources
Spatial perception of room
Concrete/Abstract applications
Reconstruct shared spaces with others
Teaching tool, to understand the technology
Build office environments where agents (employees) can be in same environment with audio passthough
Room planing, decortion, event planning
Creative process for meetings

Figure 7.13: Affinity diagram from data related to use cases from the test session.

From the interview question *"What was your reasoning to arrive at your answer?"* the participants provided a variety of reasons for their guesses. However, a notable consensus became apparent around the grouping in the middle of the room, i.e. the big table with the four chairs standing inside it, which served as the primary motivation for some of the participants but was also combined with additional factors. The Table 7.3 contains all the reasons for choosing floor plan 1, including which participants as well as number of participants. There was also one participant that said that it did not resemble any of the floor plans.

From comments made during the test as well as during the interview some information regarding elements that made it harder to understand which floor plan the 3D environment represented but also how certain aspects were perceived. There were in general a lot of comments regarding the middle table but also the tables in general. The large table in the middle was in three instances mentioned as distracting and a reason for uncertainty in the first task based on the shape of it. The geometry of the tables was recurrent during the tests, either because of them having very different scaling or because of chairs standing in them. Some of the participants also mentioned that the trashcans had different sizes as well as shapes between the two scenes. The chairs were mentioned by a few participant because they all had the same orientation, all directed towards the back wall of the room. Finally, two participants said that it was a bit difficult and hard to distinguish between floor plan 1 and 2.

When the participants were asked "*What do you think could be improved?*" the three most common answers related to the environment were to improve tables, i.e. avoid weird sizes, correct shapes and avoid overlapping with chairs, improve scaling of objects in general and to increase realism. The rotation of the chairs was also mentioned by a few people as something of importance and that it should be improved.

The tests also provided information regarding aspects that were good. Summarizing the positive feedback, multiple areas were highlighted including the design, room dimensions, positioning and resemblance. Some users found it easy to identify the correct floor plan and that the environment had good immersion and a clear neutral design. The room was mentioned to have good proportions and the elements felt plausible. Furthermore, the spaciousness made it easy and smooth to walk around. The positioning of furniture was also commented as an area that seemed to correspond with the floor plan. The chairs also received comments which included that they were accurate location wise, or that they seemed more accurate than the tables. One participant mentioned that the design of the chairs was nice, and that it was good that it was not generic.

Table 7.3: Table containing the participants reasoning behind their floor plan guess.

Reasoning	Participant	Count
Big table and chairs (middle of room)	T1, T2, T3, T4, T5, T6, T10, T11, T13	9
Easy to ruleout 3	T1, T4, T8, T13	4
Floor plan 2 & 3 were too different (more scattered)	T8, T13	2
Trashcans	T4, T5	2
Top left corner differed	T6, T11	2

Chapter 8
Discussion

8.1 Development

The code of Hydra consisted of many files. There was no official documentation of Hydra, hence, it was sometimes difficult to know how to approach Hydra and what part of Hydra to investigate when trying to understand or modify something.

The configuration of Hydra could be modified in many ways. Testing a large number of configurations of Hydra was out of the scope for this thesis, but could be interesting to investigate during further work. There are many variables that can be set to different values, affecting the scaling, separation and merging of the bounding boxes. Finding a general configuration that suits many different types of input is likely a difficult challenge. However, finding an accurate configuration for a specific scene could potentially be possible. The configuration used for the final prototype was a configuration that was tested by tweaking different values of the configuration that were thought to improve the accuracy of the bounding boxes. It was an exploratory process, not least because of the lack of documentation of Hydra. However, it could have been done more systematically, for instance by deploying an automated test that ran Hydra with a specific configuration, taking a screenshot of the resulting bounding boxes in Hydra.

There are some bottlenecks in the workflow that could be improve to further improve the performance. The objects generated in Unity were instantiated and destroyed each time a new object replaced another object, even if it was the same object according to Hydra. This approach is computationally intensive and could be improved, for instance by implementing object pooling. Another approach would be to add a method that handles updates on created objects, i.e. two topics, one for updates to existing objects and one for new objects. However, no test person complained about lag or lack of performance during the tests.

Furthermore the current setup for publishing object updates may have three possible bottlenecks. The first one would be to change our python implementations to C++. C++ is proven to have a higher performance than python in data processing and intensive loops which is included in our implementation. The second possible bottleneck could be that because object updates only filters out objects that have not changed position or size. In the case of large scans or complex environments this could prove to be a problem. The way the current filter works is by checking the that the bounding box minimum and maximum value have changed. A simple solution could be to set a limit on minimum difference between the current and new min and max value. This could improve performance and possibly avoid problems occurring when larger or more complex scans are performed. The third potential bottleneck is how the object update message was structured. Currently a list of objects are traversed and for each new or updated object a message is published with position coordinates, size, rotation and object category id. No tests have been conducted but it might be more effective to instead of publishing each time an update is identified from the input list, instead save all them in a list which is published after the input list is traversed. This could improve performance but may also add a delay or appear more choppy, all updates would be made at the same time.

Having more than one agent scanning the same environment at different parts of the environment could be an interesting experiment to investigate further. In our current implementation of the workflow, adding multiple agents scanning the same environment is in theory trivial.

Generating self-recorded input to Hydra with a camera was not implemented successfully. Some time was spent on trying to get it to work, but it was not a trivial task to manage, which was believed initially. By reading a GitHub issue regarding running Hydra with custom data, instructions were written by one of the Hydra authors, but some parts of the instructions did not include the code that was used to perform that step [11]. However, by successfully running the Project Aria Digital Twin dataset with Hydra, it could be a step in the right direction to generate self-recorded input to Hydra with a camera.

8.2 User Test Analysis

Overall, the user test sessions were generally successful, providing important and interesting information about the application and the specific parts that were of interest.

8.2.1 Participant Selection

The decision of dividing the test users in two groups was made to easier understand and discuss the results. Due to the fact that the employees were more likely to have previous knowledge or experience of Hydra and other areas that were studied in the report. The hypothesis was that we would get more data related to the underlying technology which could be interesting to compare with the other group's results. Another aspect considered was that it could affect the results which would be more discernible if separated, e.g. knowing limitations of Hydra or similar technologies could affect the user experience. Furthermore, the participants from Ericsson had more XR experience in general which also could have provided an interesting angle to investigate differences.

8.2.2 Test Setup/Session

Some problems occurred during some tests, but the most common one was connectivity issues due to a loose contact between the cable and headset. Consequently, there was an interruption around one minute between the orientation and the initiation of test one. This only happened during the first test session, i.e. the tests conducted at Ericsson. The problem was solved for the other tests with an extra Velcro strap for the other tests. It was only one instance where the connection was lost as the participant put on the headset.

Two participants experienced motion sickness during the tests. An observation made during these tests was that both of the participants used joystick locomotion for movement. Although it is out of the scope of this thesis, it is still of important consideration to further develop the application. Furthermore, it may have had an impact on the test results, e.g., impairing the participants' performance as well as negatively affecting the users experience parts tested.

Two things missed during development and pilot testing were that some objects would not be rendered or sometimes a new not before seen object appeared. The most common one was the table by the exit that sometimes never appeared. The second mistake was the scaling for the trashcans in the first test, they had a fixed scaling which made them more visible and realistic looking.

8.2.3 Test Scenario

On the whole, the test users liked the test. It was either motivated from the interest some had in Hydra and how it works or because it was a good proof of concept and others because they liked the "challenge" of the test.

8.2.4 3D Environment

The opinions regarding the environment and objects differed a lot. However, the large table in the middle was mentioned in some way by everyone. When asked what they would improve seven out of the eleven mentioned something related to improvement of the table in the middle, e.g. dividing into multiple tables or avoid overlapping of the chairs standing in the table.

Performance and Resemblance

Both groups performed similarly but some areas differed. The time it took before they guessed on which floor plan it was, all participants from the Ericsson group guessed correctly by the second guess while the other group had one participant that could not guess which one it was.

The time taken by the participants to guess which floor plan that had been recreated was initially meant to be used as an indicator or result on how well the recreated environment preserved the spatial understanding. However, during the test session we realized that some participants had interpreted the task as a challenge. As a result, the time taken to answer did not provide a reliable measurement of the participants' spatial awareness. This could have been avoided by either rethinking the type of task performed by the user or to improve instructions of the specific test.

In total ten out of the eleven participants (i.e. 91%) correctly identified floor plan 1. This result can be argued indicate that the resemblance between the environment and floor plan was good, however there are other factors to consider. The main factor to consider is if the floor plans were too different. There were participants who mentioned this. They also mentioned that it was hard to distinguish between floor plan 1 and 2 while the majority thought floor plan 3 was easy to rule out.

Although multiple participants thought that the large table in the middle of the room and the four chairs that stood inside the table were confusing and weird, it was still used either as the sole basis of the guess or in combination with something else by many of the participants. Other strategies included looking on placement and/or number of trashcans. This could further indicate that the layout of floor plans had to many differences, for example as some participants mentioned, the first floor plan had a more centered design than the other ones. If they had all had different variations of the group of furniture the middle of the first floor plan, the result could have been different.

Spatial Awareness and Perception

From the performance results, one could argue that the spatial awareness was good. However, the comments and answers from the test sessions, indicated that it might not have been the case. Multiple participants from both groups said both that it was easy but also that they were confused by the size of the middle table as mentioned previously. This could mean that the test case was flawed or to easy. One participant said that it would have been harder if floor plan 2 and 3 would have had more tables in the middle as floor plan 1.

A way to further investigate the spatial perception of the test persons would be to ask the user after the first test to mark where on the floor plan they think that they were located. Alternatively, asking them to walk to a specific position in the XR environment and then ask them to point out where on the floor plan they believe they were standing.

8.2.5 Camera

The opinions on the cameras design varied a lot, both cute, cool, nice, fun, engaging and that it indicated technology which helped visualize the purpose. But also that it looked dated, some thought it looked like an old television because of its size and model. Some also had different expectations, that it would be more human like or more like a digital camera. There were also some confusion of its purpose, that it was not completely clear at first while others thought that it effectively contributed on how the scanning worked as well as that it helped visualizing how the loop-closure worked. It is however important that users understand the camera, as one of its purpose is being a tool for affording directional information, i.e. guiding the user where to look and what part of the room that currently is scanned. Furthermore, this also served as a constraint in knowing what already had been scanned, as the camera also was meant to serve as a constraint because in our implementation the location of detected objects completely correlated to where objects was rendered in the room. However, an easy solution for this was provided by one of the participants, who suggested to add a visualization of the camera's point of view. Implementing this suggestion could be done by using the configurations used to define the scanning area, i.e. depth and width. This could further make the loop-closure more apparent.

8.2.6 Reconstruction

The reactions on the reconstruction overlapped some with the camera. Some examples on this were for instance that it was a bit hard to understand when the process had ended, both the visualization of the reconstruction and camera could be the cause of this, or presumably that none of them demonstrated this in any other way than just stopping. Furthermore, also related to the camera, one participant said that they initially felt a bit anxious about not knowing where objects would appear which suggests that indication on where objects would be rendered, also could be improved. However, there were participants who felt the opposite, that it could appear a little more explosive or that objects could be puzzled together. Related to the rendering animation of objects, it was perceived as choppy and one participant suggested to make it smoother.

The user experience varied between the participants. Some that thought it was a bit slow while others expressed it as exciting and fun. Overall the participants mentioned the experience as good. Furthermore, from the Ericsson participants, it was commented on that it aligned well with their expectations, which presumably means that some had previous knowledge of Hydra. Experiencing how the environment was reconstructed proved to provide some benefits in understanding how the objects got their scaling and appearance, which could be a good takeaway when introducing a real scanner in the process. It may for example help the user to realize that some objects need to be scanned again from a different angle. Another benefit referred to was that it gave more of a connection to the environment, which suggests that it can help with the immersion.

8.3 Ethical and Societal Aspects

As the technology implemented during the thesis may be used for real-time tracking and rendering of objects, it may also be used for real-time tracking and rendering of humans. It is important to consider the ethical aspects of using the technology, particularly if the technology would be used in a public space. If one would use the technology to scan and render another person, it would be of highest importance to have permission of doing so, stating information about how to data would be used and stored.

The societal benefit of the technology of real-time tracking and rendering of objects may for instance be its potential to enhance training and simulation experiences in industries that can benefit from realistic virtual environments based on real environments, allowing professionals to practice their skills in a controlled setting. The technology also has the potential to facilitate remote collaboration by creating immersive virtual meeting spaces. This can benefit individuals who are unable to meet physically, leading to more inclusive and interactive collaboration experiences.

8.4 Research Questions

8.4.1 RQ1

How is a user's spatial awareness affected in a virtual environment reconstructed from a real environment with Hydra?

Analyzing the results of the user tests, it was clear that the majority of the test persons could recognize the correct floor plan when they were present in the virtual environment in XR.

However, one could argue that the correct floor plan differed more from the other two floor plans, which made it easier for the test persons to separate it from the other and guess correctly. Making the floor plans more similar could have made the test regarding spatial awareness more interesting.

8.4.2 RQ2

How can real objects be tracked and rendered in real-time in the virtual environment?

During the development of the workflow, an ambition was to set up a camera for generating self-recorded input to Hydra. In order to track real objects and render them in real-time in a virtual environment in practice, a camera for generating self-record input to Hydra needed to be successfully set up. This task was not completed, hence, tracking and rendering real objects in real-time could not be tested in practice. However, the implemented workflow enables all parts of tracking and rendering objects in real-time except the self-recorded input. As the workflow was ran with pre-recorded data, and the objects from the pre-recorded data was successfully tracked and rendered in real-time, the full workflow including self-recorded data, is in theory possible to achieve, if the self-recorded data can be modified to serve as input to Hydra.

8.4.3 RQ3

How does the user perceive the virtual environment while objects are rendered in real time?

In the second part of the user tests, the users experienced the virtual environment in XR while it was being rendered in real-time. From the results, the general impression from the users was that it was cool, fun and interesting to see the objects be rendered in real-time.

Three parts of the test correlate to this: the camera, rendering and the objects. As mentioned in the discussion, the general impression of the camera was positive. It served as a tool to help the user understand what part of the room that was being scanned. However, this could have been further improved by adding visualization of the cameras point of view, due to the fact that some users did not fully understand what it was doing. Furthermore, it also made the experience more engaging and fun for some users.

The users' experience of the reconstruction process differed slightly, but in general it was perceived as interesting and somewhat explanatory to why some of the objects shapes and sizes were as they were. Some thought that the process was a bit slow and that it was a little bit hard to know when it was done. In addition to the fact that the users experienced it as cool when they saw the environment as a whole being built up step by step, some perceived the rendering animation of individual objects as choppy.

Lastly, the objects in the environment were not perceived as good. The scaling, shape and rotation were perceived as weird and something all participants in some way commented on in both of the test environments. However, as mentioned in the discussion, the negative experience of the individual objects can to some degree be reduced due to the fact that when the process is visualized it can be understood why some objects look like they do, e.g. some of the trashcans had an oval shape because it was only scanned from one angle.

8.4.4 RQ4

What is the use of 3DSG and Hydra in an XR context?

Hydra is a powerful framework that can be utilized to generate a 3DSG that holds data of a 3D environment. The data can be the basis for reconstructing a virtual environment in XR based on a real environment, as demonstrated during this thesis. If the task of setting up a camera to generate self-recorded input to Hydra can be successfully set up, Hydra may be used to reconstruct a real environment in XR in real-time.

However, the bounding boxes encapsulating the objects was not always accurate. Some objects were merged with objects next to it, creating a large bounding box of multiple objects. For some configurations, the bounding boxes was removed, making it unable to even identify some objects. To improve the bounding boxes of the 3DSG generated by Hydra, the configuration of Hydra may be experimented with in order to find a configuration that has a good trade-off between separating and merging objects. Finding a configuration of that kind would make Hydra a considerably powerful tool in terms of reconstructing a virtual environment in XR based on a real environment.

Chapter 9
Conclusion

The aim of the thesis was to explore the role of Hydra and 3DSG while reconstructing a virtual environment from a real environment in real-time, as a user experiences the virtual environment in XR. It was achieved utilizing an iterative design process, iteratively developing a prototype, incorporating areas of interest regarding the research questions of the thesis. User tests were performed in order to evaluate the prototype and answer the research questions.

From the user tests, it was evident that the participants could identify the correct floor plan. Accordingly, Hydra can be utilized in order to reconstruct a virtual environment from a real environment, achieving spatial awareness for the user present in the environment in XR. However, the distinctiveness of the correct floor plan from the alternatives might have influenced their choices.

Although the intended setup of a camera for real-time object tracking and rendering with Hydra was not achieved, the workflow that was developed successfully rendered objects in real-time using pre-recorded data. By successfully setting up a camera to generate input to Hydra together with the workflow developed during this thesis, real objects may in theory be tracked and rendered in real-time.

The user perception of experiencing the virtual environment in XR as it was being rendered in real-time varied. The camera provided a helpful visualization, but it was also confusing. The reconstruction process of objects intrigued users as a fun and interesting experience. However, some rendering of objects was perceived as choppy and incorrect.

Hydra's powerful ability to generate 3DSG in real-time may be used for reconstructing virtual environments in XR. However, challenges concerning the configuration of Hydra is of importance. Experimenting with the configuration of Hydra could significantly improve its accuracy in separating and merging objects, enhancing its ability of XR environment reconstruction.

Bibliography

[1] Sepehr Alizadehsalehi, Ahmad Hadavi, and Joseph Chuenhuei Huang. "From BIM to extended reality in AEC industry". In: *Automation in Construction* 116 (2020), p. 103254.

[2] A. Çöltekin et al. "Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions". In: *ISPRS International Journal of Geo-Information* 9.7 (2020). ISSN: 2220-9964. URL: `https://www.mdpi.com/2220-9964/9/7/439`.

[3] Microsoft Corporation. *What is the Windows Subsystem for Linux*. 2023. URL: `https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/`. Accessed: 2023-12-13.

[4] Microsoft Corporation. *What is the Windows Subsystem for Linux*. 2023. URL: `https://learn.microsoft.com/en-us/windows/wsl/about`. Accessed: 2023-12-11.

[5] Design Council. *Eleven lessons. A study of the design process*. n.d. URL: `https://www.designcouncil.org.uk/fileadmin/uploads/dc/Documents/ElevenLessons_Design_Council%2520%25282%2529.pdf`. Accessed: 2023-12-14.

[6] Design Council. *Framework for Innovation*. n.d. URL: `https://www.designcouncil.org.uk/our-resources/framework-for-innovation/`. Accessed: 2023-12-14.

[7] Design Council. *Our History*. n.d. URL: `https://www.designcouncil.org.uk/who-we-are/our-history/`. Accessed: 2023-12-14.

[8] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". In: *IEEE robotics & automation magazine* 13.2 (2006), pp. 99–110.

[9] Ericsson. *About Us*. 2024. URL: `https://www.ericsson.com/en/about-us`. Accessed: 2024-01-04.

[10] K. Holtzblatt and H. Beyer. *Contextual Design: Evolved*. BMorgan & Claypool Publishers, 2015. ISBN: 9781627055598. URL: `https://wtf.tw/ref/holtzblatt.pdf`.

[11] N. Hughes. *Hydra GitHub Repository: Issue #1*. 2023. URL: `https://github.com/MIT-SPARK/Hydra/issues/1`. Accessed: 2023-12-28.

[12] N. Hughes. *Hydra GitHub Repository: Issue #40*. 2023. URL: `https://github.com/MIT-SPARK/Hydra/issues/40#issuecomment-1775447701`. Accessed: 2023-12-25.

[13] N. Hughes, Y. Chang, and L. Carlone. "Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization". In: (2022).

[14] N. Hughes et al. "Foundations of Spatial Perception for Robotics: Hierarchical Representations and Real-time Systems". In: (2023). arXiv: `2305.07154 [cs.RO]`.

[15] Anupama K. Ingale and J. Divya Udayan. "Real-time 3D reconstruction techniques applied in dynamic scenes: A systematic literature review". In: *Computer Science Review* 39 (2021), p. 100338. ISSN: 1574-0137. URL: `https://www.sciencedirect.com/science/article/pii/S157401372030438X`.

[16] Meta. *Project Aria Glasses*. 2024. URL: `https://www.projectaria.com/glasses/`. Accessed: 2024-01-17.

[17] Inc. Meta Platforms. *Meta Quest 2*. 2023. URL: `https://www.meta.com/se/en/quest/products/quest-2/`. Accessed: 2023-12-12.

[18] United Nations. *The Sustainable Development Goals*. 2024. URL: `https://www.undp.org/sustainable-development-goals`. Accessed: 2024-01-04.

[19] X. Pan et al. *Aria Digital Twin: A New Benchmark Dataset for Egocentric 3D Machine Perception*. 2023. arXiv: `2306.06362 [cs.CV]`.

[20] J. Preece, H. Sharp, and Y. Rogers. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2015. ISBN: 9781119066002. URL: `https://books.google.se/books?id=ZC1EDwAAQBAJ`.

[21] Python. *Python Documentation*. 2023. URL: `https://docs.python.org/3/`. Accessed: 2023-12-13.

[22] M. Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[23] Intel Realsense. *Intel® RealSense™ Depth Camera D435i*. n.d. URL: `https://www.intelrealsense.com/depth-camera-d435i/`. Accessed: 2023-12-13.

[24] ROS. *ROS Documentation*. 2023. URL: `https://wiki.ros.org/ROS/CommandLineTools`. Accessed: 2023-12-13.

[25] A. Rosinol et al. "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020. URL: `https://github.com/MIT-SPARK/Kimera`.

[26] J. Rubin, D. Chisnell, and J. Spool. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley, 2011. ISBN: 9781118080405. URL: `https://books.google.se/books?id=l_e1MmVzMb0C`.

[27] M. Stickdorn and J. Schneider. *This is Service Design Thinking: Basics, Tools, Cases*. John Wiley & Sons, Inc, 2012. ISBN: 9781118156308.

[28] B. Stroustrup. *The C++ programming language*. Pearson Education, 2013.

[29] R. Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[30] Unity Technologies. *ROS TCP Connector*. 2023. URL: `https://github.com/Unity-Technologies/ROS-TCP-Connector`. Accessed: 2023-12-13.

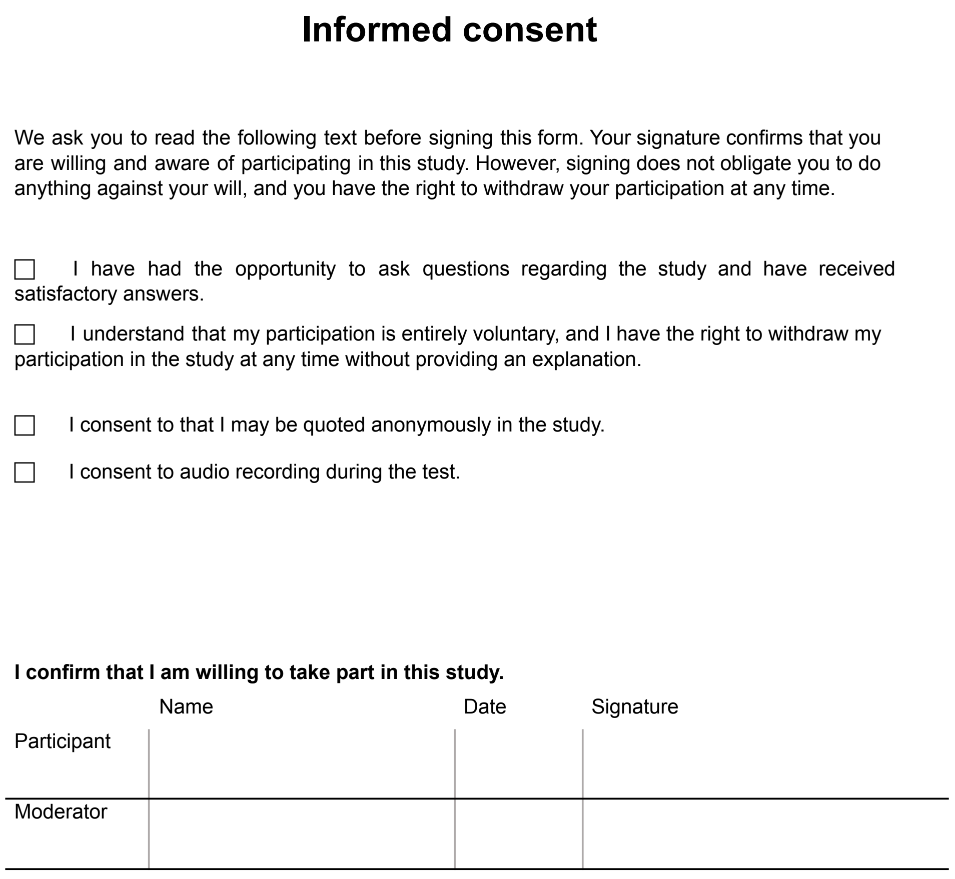
[31] Unity Technologies. *Unity User Manual*. 2023. URL: `https://docs.unity3d.com/Manual/XR.html`. Accessed: 2023-12-11.

[32] Unity. *Coding in C# in Unity For Beginners*. n.d. URL: `https://unity.com/how-to/learning-c-sharp-unity-beginners`. Accessed: 2023-12-13.

[33] S. bin Uzayr. *Mastering C#: a beginner's guide*. Taylor & Francis Group, 2022. ISBN: 9781003214779. URL: `https://books.google.se/books?id=7JvczgEACAAJ`.

Appendices

Appendix A: Informed Consent



Informed consent

We ask you to read the following text before signing this form. Your signature confirms that you are willing and aware of participating in this study. However, signing does not obligate you to do anything against your will, and you have the right to withdraw your participation at any time.

☐ I have had the opportunity to ask questions regarding the study and have received satisfactory answers.

☐ I understand that my participation is entirely voluntary, and I have the right to withdraw my participation in the study at any time without providing an explanation.

☐ I consent to that I may be quoted anonymously in the study.

☐ I consent to audio recording during the test.

I confirm that I am willing to take part in this study.

	Name	Date	Signature
Participant			
Moderator			

Appendix B: Orientation Script

Hello and welcome!

Thank you for participating in this user test for our thesis.

To ensure that all participants receive the same information, I will now read from a script.

This test is divided into **two parts**:

In the first part, you will find yourself in an office environment in virtual reality.

In the second part, you will see how the same office environment is constructed in real-time in virtual reality.

Once you have completed both tests, we will ask you some questions about your experience.

Part one:

First, you will see three blueprints of floor plans that you should try to memorize. You will then enter the virtual environment and explore it until you believe you can guess which floor plan resembles the virtual environment the most. Remember that the environment you see is recreated and may therefore differ slightly from the floor plan.

Part two:

In the second part, you will see the same virtual environment being constructed in real-time. You will see a camera moving around the room; imagine it is your colleague scanning the office environment you saw in the first part.

For both parts we encourage you to **think out loud** as much as possible, meaning: tell us what you are looking at, what you are thinking, and how you feel about it. It will help us understand your experience. And remember, we are not testing you; we are testing the technology we have developed and used.

Do you have any questions before we begin? You can always ask any question that might occur during the test.

Appendix C: Observation Protocol

Orientation:

Del 1:

Nbr times checking the the blueprints:

Timestamp, right/wrong (1/0) and guess:

Timestamp (0-120 s):	(1/0):	Blueprint nbr:

Interruptions/Cause:

Notes/Observation:

Del 2:

General notes/observations:

Appendix D: Debriefing Document

Part 1

1. On a scale from 1-5 (1: not at all, 5: identical), how well did the virtual environment resemble the floor plan?
2. On a scale from 1-5 (1: very difficult, 5: very easy), how difficult was it to understand which floor plan was correct?
3. What was your reasoning to arrive at your answer?

Part 2

4. How was the experience of seeing the environment being built in real-time?
5. How did you perceive the camera?
6. In what contexts do you think the real-time construction could be used?

Both

7. What do you think could be improved?
8. What did you like?
9. Do you have any other questions or comments?

Appendix E: Floor plans

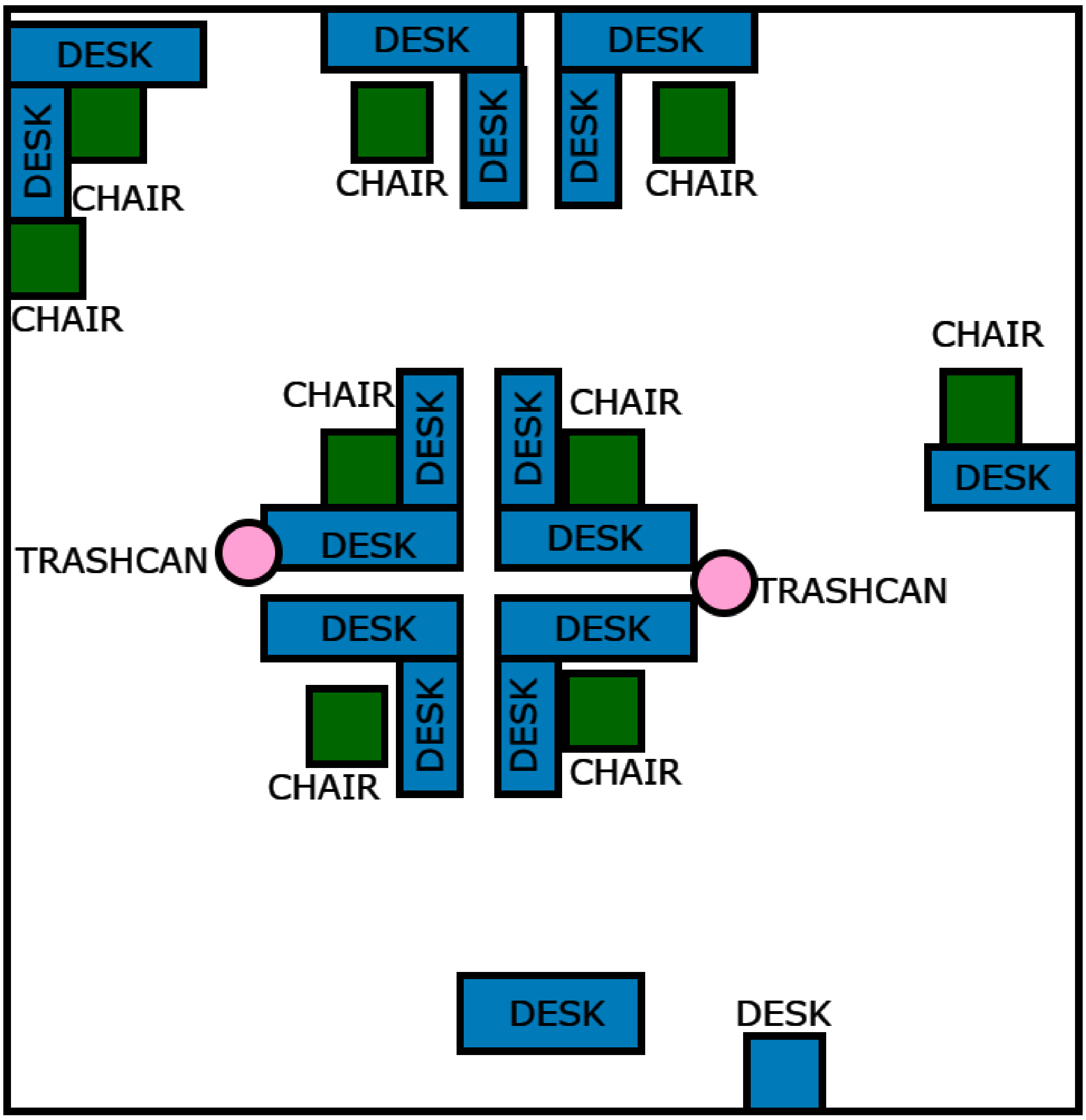


Figure E.1: Floor plan alternative 1

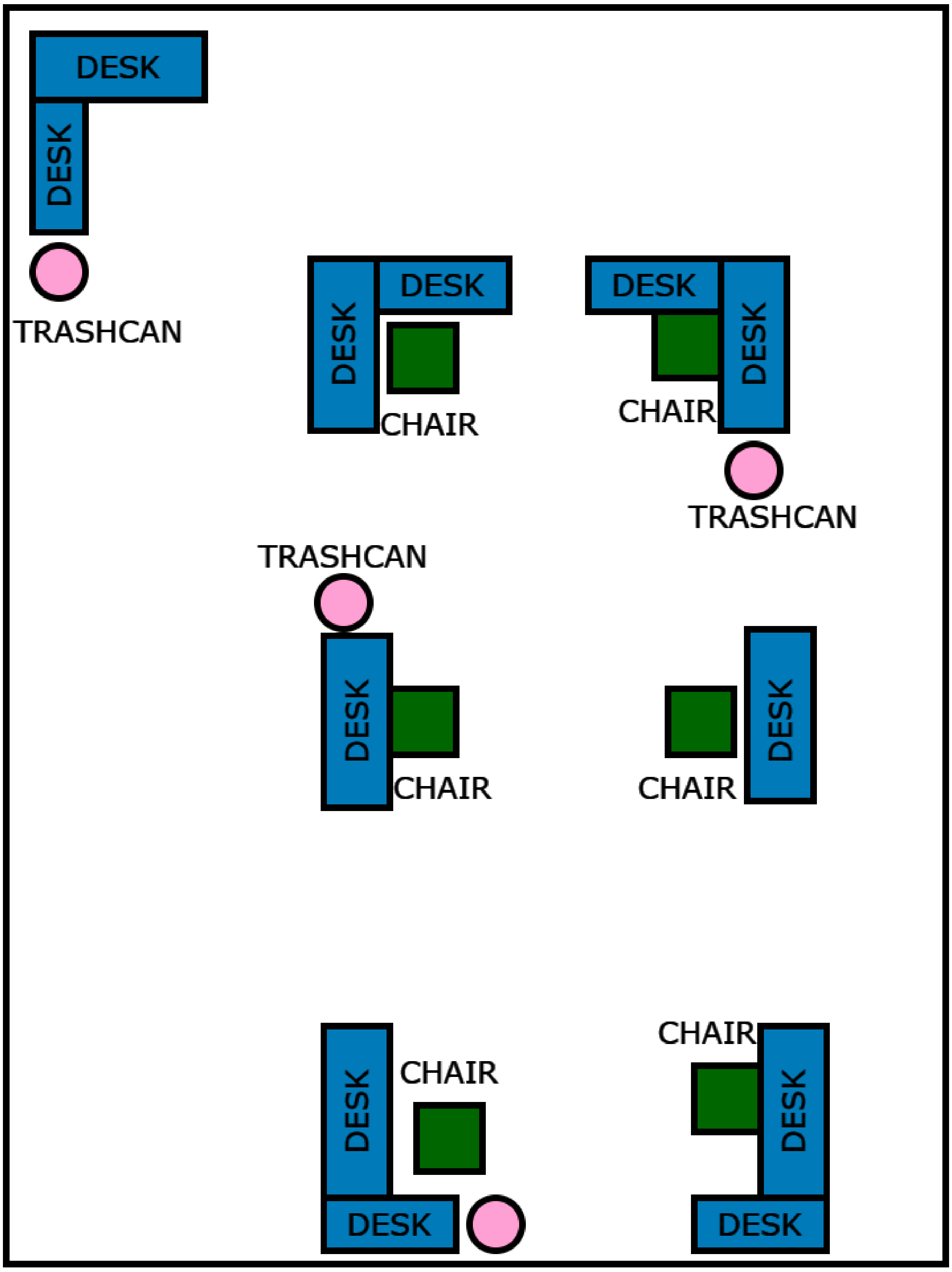


Figure E.2: Floor plan alternative 2

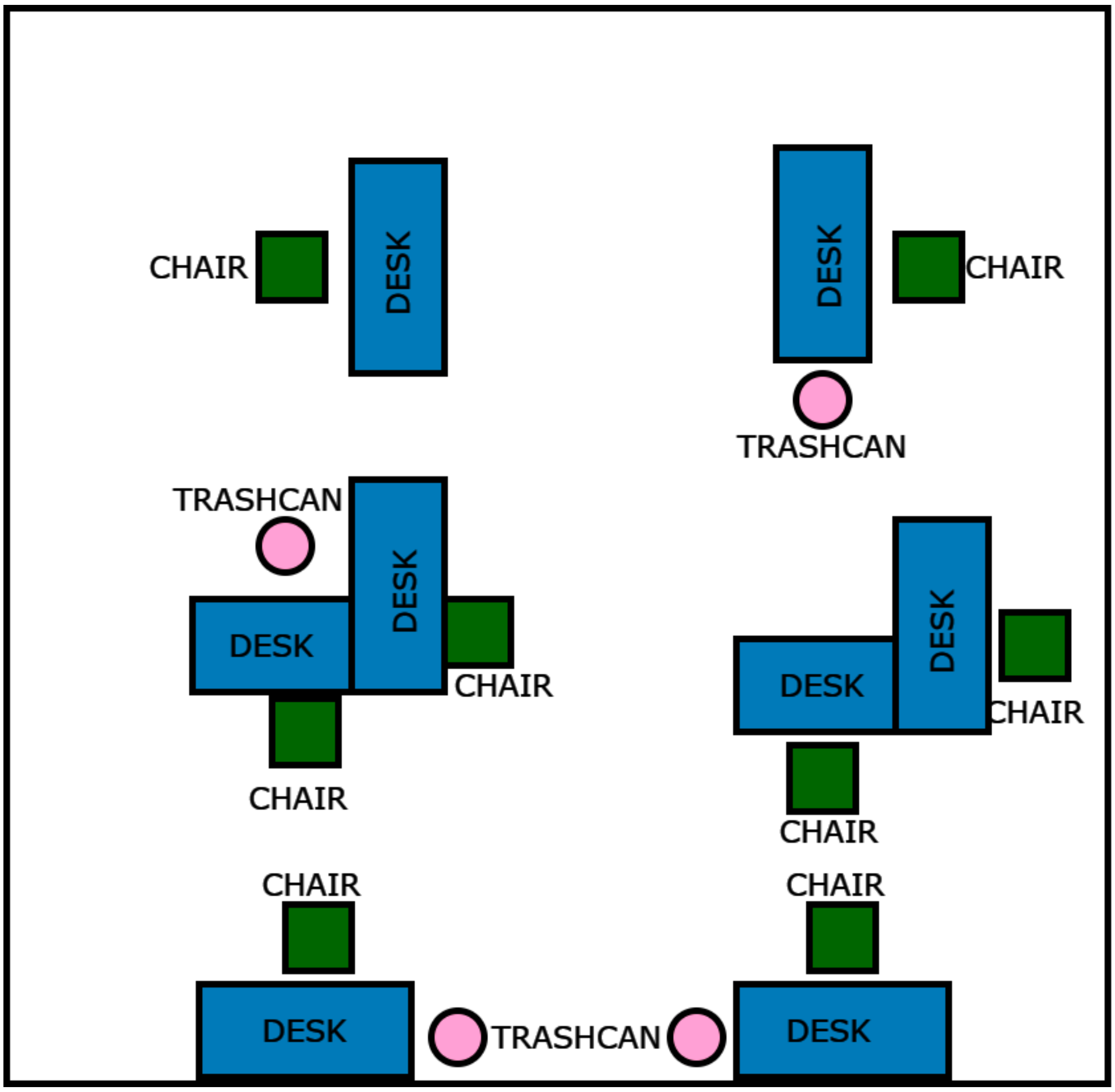
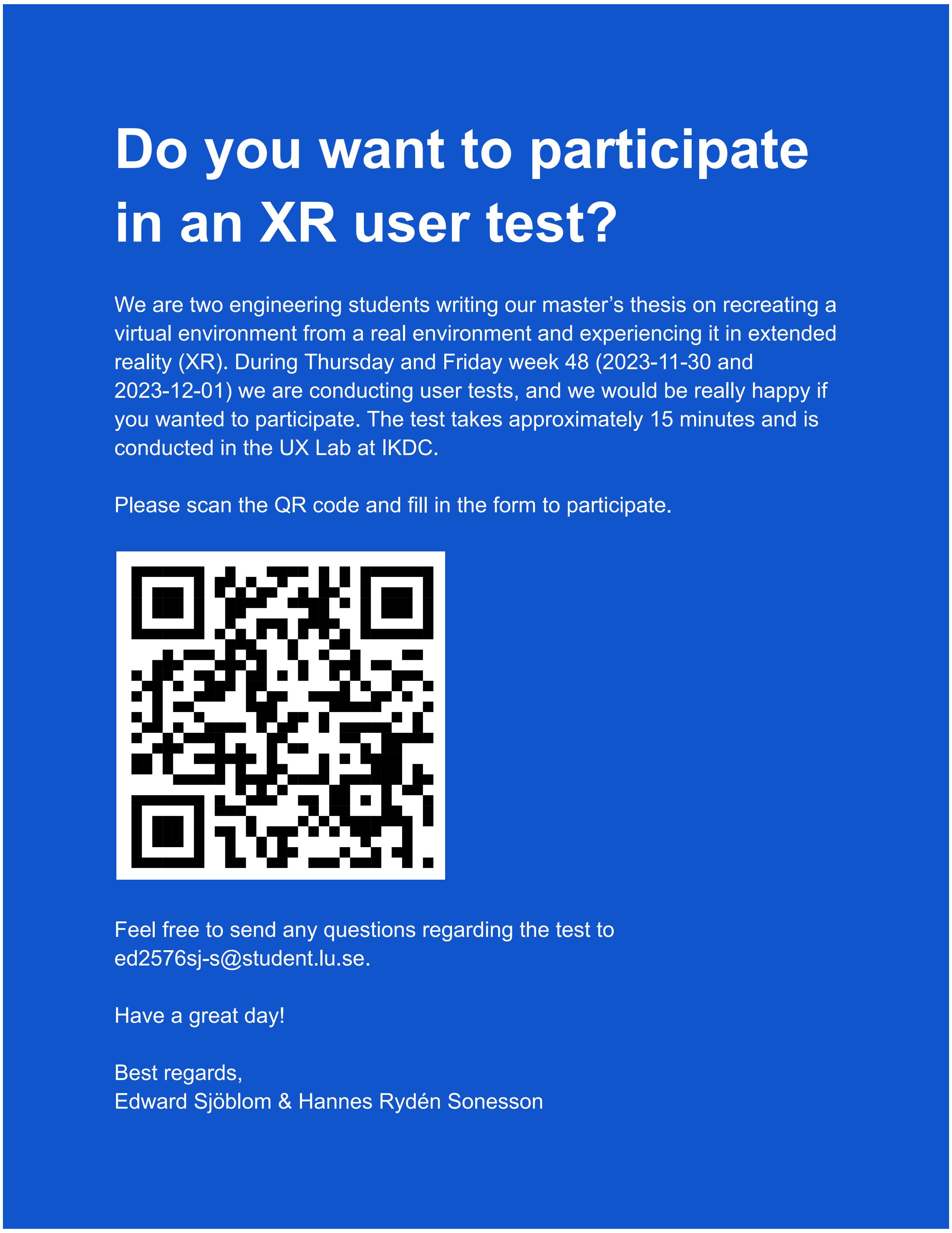


Figure E.3: Floor Plan alternative 3

Appendix F: Poster



Do you want to participate in an XR user test?

We are two engineering students writing our master's thesis on recreating a virtual environment from a real environment and experiencing it in extended reality (XR). During Thursday and Friday week 48 (2023-11-30 and 2023-12-01) we are conducting user tests, and we would be really happy if you wanted to participate. The test takes approximately 15 minutes and is conducted in the UX Lab at IKDC.

Please scan the QR code and fill in the form to participate.

Feel free to send any questions regarding the test to ed2576sj-s@student.lu.se.

Have a great day!

Best regards,
Edward Sjöblom & Hannes Rydén Sonesson

Appendix G: Form LTH

XR User test

User test regarding recreating a virtual environment from a real environment and experiencing it in extended reality (XR). Location: The UX Lab at IKDC in Lund.

* Indicates required question

1. Name: *

2. Age: *

3. Gender: *

Mark only one oval.

- ◯ Male
- ◯ Female
- ◯ Non-binary
- ◯ Other: ______

4. Experience with XR: *

Mark only one oval.

	0	1	2	3	4	5	
Nev	◯	◯	◯	◯	◯	◯	Professional

5. Preferred time slot: *

Mark only one oval.

- ◯ Thursday 2023-11-30: 09.00
- ◯ Thursday 2023-11-30: 09.30
- ◯ Thursday 2023-11-30: 11.30
- ◯ Thursday 2023-11-30: 13.00
- ◯ Thursday 2023-11-30: 14.30
- ◯ Thursday 2023-11-30: 16.00
- ◯ Thursday 2023-11-30: 16.30
- ◯ Friday 2023-12-01: 09.00
- ◯ Friday 2023-12-01: 10.00
- ◯ Friday 2023-12-01: 10.30
- ◯ Friday 2023-12-01: 11.30
- ◯ Friday 2023-12-01: 13.00
- ◯ Friday 2023-12-01: 13.30
- ◯ Friday 2023-12-01: 14.30
- ◯ Friday 2023-12-01: 15.00
- ◯ Friday 2023-12-01: 15.30
- ◯ Friday 2023-12-01: 16.00
- ◯ Friday 2023-12-01: 16.30

6. Email address: *

Appendix H: Form Ericsson

XR User test

User test regarding recreating a virtual environment from a real environment and experiencing it in extended reality (XR). Location: Komodo at the Ericsson office in Lund.

* Indicates required question

1. Name: *

____

2. Age: *

____

3. Gender: *

Mark only one oval.

◯ Male
◯ Female
◯ Non-binary
◯ Other: _______________

4. Experience with XR: *

Mark only one oval.

	0	1	2	3	4	5	
Nev	◯	◯	◯	◯	◯	◯	Professional

5. Preferred time slot: *

Mark only one oval.

◯ Tuesday 2023-12-05: 09.00

6. Email address: *

____

Appendix I: Raw Affinity Data

Asks to see floor plan again	Asks about floor plan and the instructions	Guess correct first time, checks blueprint after and confirms the guess	4 chairs inside the middle desk	The battery of the HMD ran out, so had to charge the HMD for 20 minutes as the test person went away. The test person looked at the floor plans before going away	Nice that floor plans are different. That they all have clear characteristics that separate them	Asked how the scanning was done	Asked if you will be able to move around or if you will be a part of the camera in part 2	A bit uncertain at first but further investigating increased decision	Notices a chunk of objects in the middle, which was only in floor plan 1	Tried to grab stuff
Asks if the entrance is part of the floor plan	Guess wrong first but change to the correct after seeing the floor plan again (takes of HMD)	Had about 1 minute initial interruption because of HMD connection complications	Trashcans by the middle table	The chairs are inside the table	Notice big table in the middle, bases for guess on floor plan 1	Did not think the environment resembled any floor plan	Said that it sounds like the tv-series “Severence”	Thought it was floor plan 3 because of the number of tables, but had mixed up the numbers and actually meant number 1	Noticed the number of trashcans	Think floor plan 1 had a more centered design than the other ones. Which ruled out the other floor plans.
Asks if direction of chair matter		Had some connection problems with the hand controllers		The chairs are thin and tall	Comments that the desks by the walls are similar to floor plan 2	Thought the environment was messy. Objects were inside each other, the chairs are rotated wrongly, everything looks randomly scaled	Commented that the furniture in the middle of floor plan 1 probably will be easy to recognize. Floor plan 2 and 3 are more similar		Tried to understand where in the floor plan he was located	Would have been good if you could interact with object, e.g. sit and spin a chair or open a drawer.
The middle big table can be either floor plan 1 or 2		A lot of chairs in the table			Likes that the floor plans separate objects by color.	Looked for L-shaped tables	Had some problems with the HMD. Had to reboot the HMD and the Oculus app. Took 2-3 minutes			Relaxing walking around
Confusing with the big middle table, should be a cross		Asks if this is the input to Hydra				Mostly looked for big formations	Decided quickly that it was floor plan 1 and kept that guess			
Can’t guess floor plan from number of tables										
Chairs stands in the tables.										

Camera is funny	Motion sickness	Does the camera follow something from Hydra?	Nice camera	Thought the camera would be more human-like	Nice to see how the objects grow (appears)	Asked if it was the same environment after 10 seconds	Did some boxing towards the camera	Could see it was the same environment as in the first test.	The camera was cool	Asks if the camera scans the room.
Trashcans have weird shapes	Uses left joystick	Asks about if that the camera is still means that Hydra cannot handle more input	Cool	The trashcans have shrieked	At first a bit anxious to be in the way of objects that spawn, but could understand that it wouldn’t matter.	Asked if the lamps are included	Was not totally clear if it still is rendering		The camera was big	Very cool to see the room (obejcts) being built
Height of objects differs from room in test 1	Says it is the same floor plan as the first one	Feels good and plausible	Asks if it was meant to be L shaped tables	The weird side table is gone or has moved, I do not know	Don’t know if she likes it more than the finished environment, but appreciate both.	Wondered why the chairs were scaled differently	Notices that it is also is a lump of furniture in the middle		I think it was another desk in part 1	Would be interesting if something would be shown on a painting
Chairs seems to be more accurate than table (position)		Wanted to know how the textures of the input objects looked	Small trashcan in comparison to room 1	The ceiling windows appeared a little later, did not feel that they were there before	Likes how the ceiling windows appeared, or if it was ceiling lights, thigh they where nice.	Wanted the furniture to be modular, but different sizes of tables for example	The chairs are in the table		Does not recognize the trashcan in the top of the room	Nice design on the chairs.
		Looks weird with the table and the chairs	Oval trashcan one side of the big table	Got dizzy	Likes that there was an avatar (camera) that indicated were objects would appear, by it’s moment and rotation.	Interesting with floor plans that you have a perspective from above and can create formations from that	It feels like a cult because the chairs are rotated towards the same direction		Feels like something was in the beginning or the end of the room that is not there anymore	Asks if it was a cinema or an office, because of all the chairs pointing to a wall.
		The table looks big	Small table in the end are missing		Cute figure		Plays around a lot in the environment and is teleporting a lot			Because of all the chairs pointing in the same direction, it felt like something would happen on the wall if you sat down on the chairs.
			One chair is missing							

Confusing with big table in the middle	Thought it was floor plan 2 because of the desks along the wall but the big middle table changed the answer to floor plan 1	Can see which floor plan it is but it does not look like a real environment because of the chairs in the table and that the geometry of the tables does not resemble the floor plan	The environment resemble the floor plan except the desks	Depends on the purpose. If you should be able to interact with the objects or if you should understand	It was clear that it was the first floor plan because it was a lot of things in the middle of it.	Understood how they resembled each other now, but hard to perceive	Feels like the furniture are in the correct positions, but the tables in the middle became one big table instead	It was the tables in the middle that indicated floor plan 1,	Was very clear that it was floor plan 1 because of the furniture in the middle	Became a bit unsure which floor plan it was after looking at them again
Hard to distinct between floor plan 1 and 2		Looked at the big table in the middle to see that it was floor plan 1	The desks have different sizes	The reasoning was that there were two trashcans and that the table was in the center	It differed in the top left corner of the floor plan.	Looked for L-formed desks, so got distracted by the large table	Initially mostly focused on the chairs and the trashcans	Uncertainty because of big table in VR but cross shape in floor plan.	Also knew that there was a desk in the top left corner	It’s not identical but the middle table was pretty similar and the chairs in the middle spot on. The other parts he couldn’t recognize.
Easy to rule out floor plan 3			Used the middle table and chairs and may be trashcan to guess floor plan.	The chairs of floor plan 2 would be rotated in another direction			Got a little competitive and did not care so much about the rest			The other floor plans was more scattered.
			Easy to rule out floor plan 3				Kind of similar to the floor plan			
			A bit difficulty to differ between floor plan 1 and 2				Felt that objects would be rotated to different directions			
							Was very easy to identify that it was floor plan 1			
							If floor plan 2 and 3 had a big table in the middle it would have been harder to identify			

Cool to see environment be built	Was expecting what was shown and it made sense that objects occurred when the camera was moving around the room	Interesting to see the environment being built in real time since I have started researching Hydra	The reconstruction was pretty good but could be faster	Did not know how to know when I was done. Did not know when the camera was done	Fun to see. Good that to have the avatar to follow.	Interesting, gave more understanding of how the environment is represented.	Thought I would guess on all floor plans, so thought something was wrong first	The real-time recreation made it easier to see/ understand where objects would appear which floor plan it was.	Exciting and fun to see the environment be built in real-time	It was interesting going from an empty room to things starting to appear.
Great future outcomes		Aligned well with my expectations		The trashcans and the tables were different in part 1 and part 2	It indicates technology and didn’t feel weird.	Could understand how the different scalings occurred	Did not think about anything in particular		I am a visual person, so would have like it a bit more graphical	Could have been more explosive how the objects appeared. Or if the objects been more puzzled together.
Can be good as a teaching tool for how scanning works.		Asks about the speed of the rendering in relation to rviz			Gave more ownership of the building process, in comparison to the first test.	Got the feeling that something was scanned	Thought it was a bit slow, did not catch up to my speed		It was a bit choppy, could have smoother	
							Felt that the camera had a connection to what was being rendered			

83

I. Raw Affinity Data

Retro camera	The camera was a bit weird at first but made sense	A little slow sometimes	The camera was graphic	The camera looked like a pre macintosh, did not immediately recognize it as a camera	The camera was helping with knowing where to look.	Contributed with understanding of how the information was processed	It was bigger that I thought	The camera provide a good indication on where you are and what has been scanned.	Did not think it would be that big	The camera was cool
Nice camera but proportion to objects were off.	Good to show camera when scanning.	Was slightly skeptical to the camera		Was not totally clear that the rendering was through the camera		Big camera, could be visualized in another way	Looked like an old TV		Looked like a TV	The camera was a little bit hard to understand
Rotation of camera good	May be good to have an avatar/person if the scene is more interactable.	Liked that improvements that were made to objects after the camera had looked away was demonstrated, so that loop-closures could be visualized				Good that it moved around	Thought it would be more like a small film camera		Used to cameras having a specific appearance	Unclear how the objects appeared in relation to the camera.
Could be good to visualize cameras field of view							Easy to see, which was good		Fun with a friend walking around	

Could be used for anything	Good for AR game if you use an environment from a friend/unknown environment	Some application with digital twins	Thinks it could be used with multiple devices that build the scene from multiple places	Virtual tourism	Could be used in a creative process. When having online meetings to increase the creativity	-	To save resources, so you can show only what needs to be showed	It can be used if there is a big environment with alot of detail (e.g. obstacles), feedback on what has been scanned and not.	Real-time could be used for big events	Office environment, to being able to build the office depending on the days need, number of employes and so on.
Especially good for creating digital twins	Could be good to see the real-time rendering process	Same virtual and physical environment with others, so that people can collaborate	Could be more abstract, e.g. boxes are spawned labeled and when done transformed to real object.	Maybe don't show what has and has not been rendered before, like a game with a map where the environment is not fully discovered	When planing/decorating a room, helps with how the objects relate to each other.		Depends on the purpose		Real-time could also be used if many guests are attending something and you need to test where to place the furniture	
Architecture	Cool to see it, be can't come up with any applications		To create a digital twin. E.g. being able to sit at home but in the office space with colleagues, with passthrough sound for communication and may be include a sound profile of the office and distance based sound	Would be nice to be able to distinguish between what has and has not been rendered			Maybe a horror game with a bad line of sight		Everything does not need to be real-time	
To reduce large scans				Hard to know where it would render						

Improve grouping of tables	Realism could be improved	Improve the structure of the big middle table	The graphics could be improved	Important that the scaling and rotation is correct, not so important that it looks nice	Unsure of what to improve	The visualization in terms of position and scaling. You should work with the visual parts, should be same scaling of objects that are the same type	Fix the tables so they look like the blueprint	The graphics could be improved	The tables in the middle should have another shape	Could be improved with added interaction, making it more exciting.
Separate tables	Not have chairs in desks	Improve the scaling	Teleportation was good but continued movement was very slow.	The chairs in the table	Likes that the maps have color differences.	Avoid having objects inside of each other	Faster rendering when the camera is scanning	One improvement would be to be in first person view of the camera, too easily see what has been scanned. Easy to miss when moving and looking around freely.	The trashcans had different scalings in part 1 and part 2	
Have more precise space?	Trashcans were hard to see, could be improved	Improve the physical relationships (chairs inside of tables)		Prioritizing of objects, trashcans less important than tables	Doen't think VR needs to reflect the exact environment, so could have made it more fun.	The chairs were all rotated to the same direction	Totally depends on the purpose	The scales could be improved. Felt like the room should have been bigger.		
Visualize cameras field of view					Make the trashcans more clearly	Could have been more of a visual abstraction. Maybe a surface with a label or an icon indicating what type of object it is				
Use more complex space (environment)					Colors would make it more interesting, and if used a little could contribute to more creativity and productivity.					
Run with real life scanner					Make the floor plans with more detail, where windows and doorways are located for example					

Nice to see and understand how the recreation and scanner works	Liked the comfort (immersion), felt like being inside the virtual environment	Good proof of concept	Likes VR	Liked the teleportation transportation method	Likes that there where a lot of space to move around in.	The combination of scanning the env ironment and VR is interesting	Looked pretty good	The visualization was good	Smooth to move around in the environment	Liked that you started in the empty corridor and entered the room.
Likes to see how objects are merged/grouped together (when it should happen)		Rendering		Liked the proportions as they seemed plausible	Surprised when accidentally teleported into a wall.	I really got the feeling that I was in a real room	The individual components looked good	The scanning was good, that you could see what had been scanned.	Was pretty easy to identify	Liked the ceiling windows
Likes rotation and location of objects		The things I did not complain about		Liked the idea	Clear design of the room and that nothing draw the attention to much.	The fundamental concept is really interesting	I liked the color scheme in general		Easy to use	Liked the design of the chairs, and that they weren't generic
Likes that it runs in real time						Could be used for the spatial perception of a room	Could put stuff on the walls to make it feel less like a test environment			Liked the camera because something happened.
Likes that it could decrease time to scan and recreate an environment.							It was smooth to move around			It was an exciting concept to try to understand which floor plan the environment represented.
							I liked the camera			

	Nice experience, curious about scene construction scenes. Good application for that								Fun to do it in VR	
	Asks about construction of 3DSG and if agents position is included in the 3DSG									