Collaborative Augmented Reality Platform: A Feasibility Study

Mikkeline Elleby[1]

Supervisor: Prof. Lalanne, Dr. Simon Ruffieux and Yong-Joon Thoo

March 10, 2023

Department of Informatics - Bachelor Thesis

Département d'Informatique - Departement für Informatik • Université de Fribourg - Universität Freiburg • Boulevard de Pérolles 90 • 1700 Fribourg • Switzerland

phone +41 (26) 300 84 65 fax +41 (26) 300 97 31 Diuf-secr-pe@unifr.ch http://diuf.unifr.ch

Accepted for: *Human-IST Research Center Documents Factory*, 1(1):1–2, February 2006.

[1] mikkeline.elleby@unifr.ch, Bachelor project with HUMAN-IST, DIUF, University of Fribourg

Abstract

This work explores the concepts of multi-user capabilities in Microsoft Hololens devices to fill a significant gap in the literature by addressing and empirically measuring the limitations and feasibility of a collaborative solution in MR, with additional considerations for LVR. A comprehensive analysis of technical solutions for implementing collaborative features was conducted, identifying Microsoft Shared Experiences as the most suitable solution. A proof-of-concept implementation utilizing the PUN server for graphics synchronization and an Azure Spatial Anchor for a common coordinate system was developed. An experimental design and evaluation was carried out to asses the limitations and feasibility of the implementation. In this context, we have focused on the position of a virtual object shared between two users, and compared the robustness of its position. Results showed accurate recognition of the object's position while highlighting challenges related to the shared coordinate system and displaying correct depths. This study shows promising potential for collaborative application in MR.

Keywords: Human-IST Research Institute, Microsoft Hololens 2, MRTK, Azure Spatial Anchors, PUN, Collaboration, Mixed Reality

1 Introduction

Vision is a critical sense that allows us to perceive and understand the world around us, communicate effectively, and carry out daily tasks [12]. However, visual impairments can greatly impact a person's ability to function optimally. According to the World Health Organization, approximately 2.2 billion people worldwide have some form of visual impairment, with tens of millions experiencing severe visual impairment or blindness [12].

Low vision is a type of visual impairment that cannot be corrected with lenses or standard treatment. It can be caused by various diseases or injuries to the eye or brain. To a certain extent, the quality of the residual vision can be restored or maintained through consistent Low Vision Rehabilitation (LVR), which uses visual training exercises to promote visual learning and perceptual development [5]. While the traditional approaches have proven effective, they have limitations such as repetitiveness, lack of immersion and disregard for real-life challenges faced by patients. Therefore in recent years, researchers have tried to use the new advancement in technology to overcome these limitations.

This research is part of a larger project led by the Humanist Group at the University of Fribourg, which seeks to explore innovative technologies for LVR. This paper aims to investigate the available methods, plug-ins, and toolkits for implementing collaborative AR, implement a technological proof-of-concept, and evaluate the limitations of collaborative AR to determine its applicability for further research.

The idea of incorporating collaborative rehabilitation tasks has been raised multiple times by participants in various exploratory studies [8]. It has been suggested that such activities could enhance motivation and overall therapeutic outcomes. Several technical solutions, more or less recent, already enable the implementation of collaborative aspects for the HoloLens 2. However, these technical solutions are not always up to date and, therefore, not compatible. They are often poorly documented and do not work with "out of the box" applications. Furthermore, the precision, robustness, and limitations of these solutions are inadequately documented.

In order to design collaborative LVR tasks effectively, it is crucial to develop an initial functional prototype (proof-of-concept) and evaluate its potential limitations. The objectives of this work encompass several key aspects.

Firstly, a thorough analysis of technical solutions for implementing collaborative features was conducted. This involved examining various options and determining the most suitable solution.

Secondly, a proof-of-concept was implemented using the selected solution, showcasing its feasibility and functionality. Additionally, potential limitations and constraints were identified during the implementation process, leading to valuable recommendations for optimal usage. To evaluate these limitations, a user study was conducted to gather empirical data and assess the performance of the implemented solution.

The results obtained from this work have broader implications for future developments in two primary areas. First of all, the prototype developed should serve as a solid foundation for further refinement and expansion, facilitating the seamless implementation of future collaborative tasks. Secondly, the identified limitations and recommendations provide valuable guidance for designing collaborative tasks that are tailored to specific constraints, ensuring optimized performance and user experience

1.1 State-of-the-Art

In recent years, there has been a growing interest in the field of low vision rehabilitation, driven by advancements in immersive technologies. Conventionally, rehabilitation methods for individuals with visual impairments have relied on traditional approaches, including the use of optical aids, specialized training with therapists, and visual training on physical boards or computer screens. While these methods have proven effective, they have limitations, including repetitiveness, lack of immersion, and disregard for real-life challenges faced by patients. The juxtaposition of these methods is shown in Figure 1.

To overcome these issues, researchers are currently investigating the potential benefits of utilizing assistive and accessible technologies to enhance rehabilitation therapy and support individuals in various tasks. Furthermore, these technologies are also being explored as a means to simulate diverse real-world environments, which can aid in the training of orientation and mobility skills. One promising direction is the integration of virtual reality (VR) and augmented reality (AR). These solutions offer new possibilities for enhancing visual perception, improving functional abilities, and enabling adaptive rehabilitation strategies. By leveraging VR/AR, individuals with low vision can engage in experiences and training scenarios.

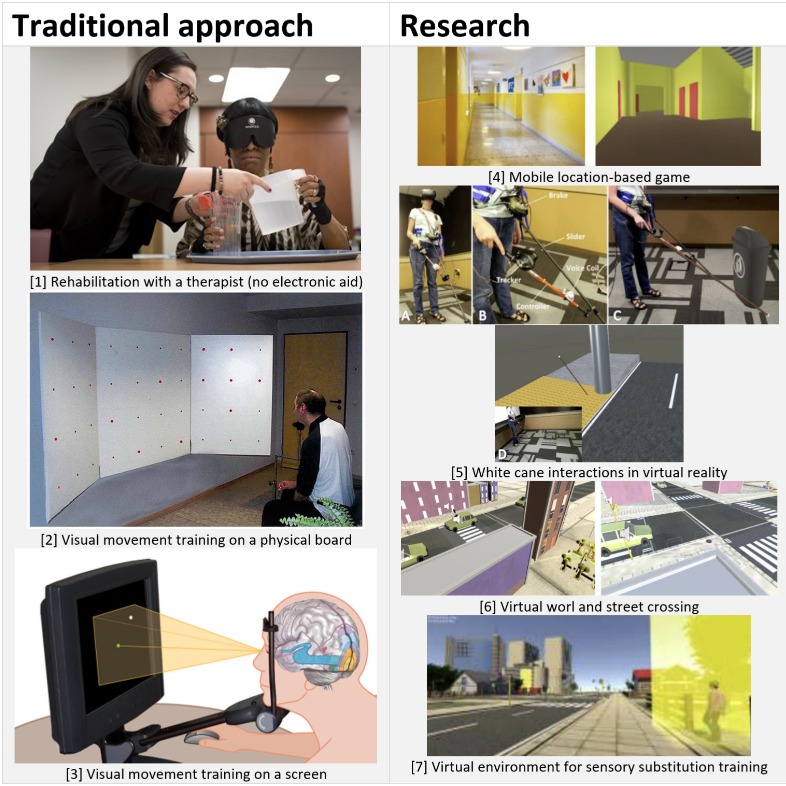
Such a solution takes the form of a scavenger hunt-like location-based game designed to support orientation and mobility training. Research has indicated that this type of training is more motivating compared to traditional approaches. Games, in general, are known to be enjoyable, which is why leveraging game-based approaches for orientation and mobility training is believed to enhance motivation and, consequently, learning outcomes. By incorporating game elements into training, individuals with visual impairments can engage in an interactive and stimulating learning experience, fostering motivation and facilitating skill development [8].

Another solution in the fields of low vision rehabilitation is Canetroller, which leverages VR technology. Canetroller is a haptic cane controller designed to replicate the interactions experienced with a white cane. This innovative device enables individuals with visual impairments to navigate and explore virtual environments by utilizing their cane skills in the virtual world. This enables people with visual impairments to access virtual reality, as they do not require visual feedback [13].

Furthermore, there have been attempts to improve the interaction skills of individuals with low vision in their surroundings. One such solution is called X-Road, which utilizes VR glasses for orientation and mobility training. Its primary focus is to teach individuals with visual impairments how to navigate the physical world, including the crucial task of crossing roads safely. The relevance of this solution lies in the fact that real-life training for such tasks can be difficult and dangerous due to conditions such as traffic and weather. By providing a simulated environment, X-Road offers a safe and controlled platform for individuals to learn and practice essential skills, ultimately improving their independence and safety in real-world scenarios [10].

While these experiments are interesting, they often do not extensively explore the use of augmented reality, especially optical see-through smart (OST) glasses. OST offer a promising solution to the challenges encountered by individuals undergoing low vision rehabilitation. As it can reduce the risks of disorientation and collision commonly experienced with virtual reality glasses. The use of OST smart glasses allows users to maintain situational awareness and engage in face-to-face communication with others in the room, which helps alleviate social isolation [14]. The device's design enables users to move in any direction and interact naturally with AR content as naturally as with physical objects, promoting effective learning and technology adoption. All of these factors highlight the opportunity to engage multiple senses when wearing the glasses, which is beneficial for therapy. Additionally, AR content can be tailored to be interactive and engaging, making it a valuable tool for enhancing rehabilitation outcomes by increasing patient motivation to engage in the different activities [4]. This results in the medium being an interesting choice for LVR.

Furthermore, patients have expressed interest in the ability to engage in collaborative activities, playing with or against others, or collaborating with them, as this can significantly enhance the



1

Figure 1: Different approaches to LVR

motivation and effectiveness of rehabilitation tasks. AR provides a potential solution to fulfill this need. Through AR, multiple headsets can connect and interact with the same virtual objects, facilitating shared experiences and collaborative interactions. However, such collaborative solutions within the realm of LVR have remained relatively unexplored.
There exist several examples of the use of AR to support face-to-face collaboration. One notable system, the Transvision system, allows multiple users to share computer-generated graphics on a table, enabling collaborative engagement [9]. More recent advancements have considered the use of AR in dynamic emergency response tasks and even within board games [6], where users interact with physical game pieces on a tabletop using handheld devices [2]. Similarly, ARVita facilitates multiple users wearing head-mounted displays (HMDs) to observe and interact with dynamic visual simulations of engineering processes around a shared table [1]. While numerous examples of non-remote collaboration in AR exist, few studies have specifically measured the precision and effectiveness of such interactions. To address this gap, we will develop a proof of concept in order to evaluate the feasibility and limitations of implementing collaborative AR experiences within LVR tasks.

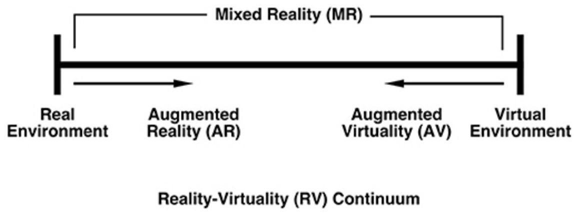
2 Materials and Methods

This chapter provides an overview of the technology employed in the project. It focuses on the chosen medium for the implementation, the device on which the application is developed, and the framework utilized to achieve the desired functionalities.

2.1 Mixed reality and the Microsoft Hololens

In recent years, virtual reality (VR), augmented reality (AR), and mixed reality (MR) have gained significant popularity as technologies that enhance users' perception of their environment. While

all three technologies contribute to the user's sense of experience and reality, they operate in distinct ways.
VR immerses users in a completely virtual environment, in such a way that their sight is completely blocked by a screen and the interaction with the user's surroundings is purely virtual. AR, on the other hand, keeps the users in their surrounding. This means that a user is able to view the world around him and the virtual elements are displayed therein. However the object, that are displayed, are simply superposed into the real world. This technology does not have the capabilities to manage the relationship between virtual and real objects.[11].



2

Figure 2: Milgram and Kishino's reality-virtuality continuum

To address this challenge, MR combines aspects of both VR and AR. MR technology tackles the issue by performing real-time spatial mapping, utilizing sensors embedded in the devices to capture and interpret the user's surroundings. This allows for the transformation of the physical world into a virtual representation in real-time. By doing so, MR technology enables the accurate placement of virtual objects based on depth, spatial persistence, and perspective, effectively integrating them into the user's perceived reality.
But it has to be noted that this definition of MR can be debated, as MR can take many forms. It is a spectrum going from real elements to virtual elements as shown in the diagram by Milgram and Kishino in Figure 2. The MR that is used in this task could be placed near the beginning of the spectrum in the above continuum, close to the real environment.
The Microsoft Hololens 2, developed by Microsoft in 2019 is an example of an optical see-through MR device [3]. It runs on the Windows operating system. It consists of 3D perspective holographic glasses equipped with a central processing unit, graphics processing unit, and holographic processing unit. These components enable real-time spatial mapping and processing, seamlessly integrating virtual elements into the user's physical environment by projecting light onto their natural vision.
Furthermore, the Hololens outperforms traditional AR devices and stands as a strong contender in the smart glasses market due to its advanced functionalities. These include stereoscopic 3D displays, gaze design, gesture design, and spatial sound design, enhancing the overall user experience and interaction with virtual content [7]. This results in an intuitive and immersive device. Therefore, it was chosen for the project and the implementation of the task.


3

Figure 3: The Microsoft Hololens 2

2.2 Unity and the Mixed Reality toolkit

Unity is a versatile game engine created by Unity Technologies, which was initially announced and released in 2005. It has since become one of the most widely used and respected game development platforms in the industry. Known for its user-friendly interface and extensive features, Unity has gathered a reputation for being accessible to both new and experienced developers alike. Unity can create games for various platforms. One of the notable strengths of Unity is its ability to support both three-dimensional (3D) and two-dimensional (2D) game development. This makes the development easier as one can leverage its powerful rendering engine and physics system. Furthermore, Unity has an active and supportive community of developers who contribute to its vast library of plugins, assets, and resources.

The Mixed Reality Toolkit (MRTK) framework is an essential part of Mixed Reality development. It offers a comprehensive set of components and features designed to facilitate the creation of cross-platform MR apps in Unity. The MRTK supports a wide range of devices and facilitates rapid prototyping. Additionally, as a community resource overseen by Microsoft, the MRTK is constantly evolving with the addition of new features. It consists of three repositories on GitHub, each with its own issues section where developers can seek assistance with implementing from tutorials and address any challenges they encounter.

For this project, we utilized the MixedRealityToolkit-Unity repository, which includes Unity-specific components. The "regular" MixedRealityToolkit, on the other hand, serves as a generalized version that contains the core C++ and HTML code based upon which many of the Unity toolkit features are built or wrapped. By supporting the Unity game engine and utilizing its various features, the development become more intuitive and visual.

3 Analysis

In this chapter, we will examine different methods, plug-ins, and toolkits that can be used to implement collaborative augmented reality (AR). The focus will be on solutions that incorporate the concepts and materials presented in the previous chapter. Especially the Microsoft Hololens 2 as this is the resource that is employed by other researchers that are part of the HumanIST group. The objective is to explore and evaluate these options to determine their suitability for enabling collaborative AR experiences.

3.1 Search method

The research methodology employed for this study encompassed several approaches. The initial step involved conducting an extensive literature review to identify relevant papers and articles regarding collaboration in augmented reality (AR). The research was conducted with the help of the library of the university, many papers were accessible from its website. Then these sources were carefully evaluated to determine their applicability and relevance to the current implementation.

Furthermore, the investigation aimed at examining various methods utilized for creating a shared environment in AR and achieving synchronization among multiple users. This involved analyzing existing techniques, frameworks, and tools employed in the field.

To complement the literature review, online resources and forums dedicated to AR, particularly those provided by Unity and Microsoft, were utilized. These platforms offered valuable insights, discussions, and community interactions related to AR development, collaboration, and best practices.

3.2 Software Architecture for collaboration

Achieving multi-user capabilities can be solve via several approaches, but one stands out as the most common. Our analysis has identified two fundamental aspects that are essential for ensuring the proper functionality of the system. These aspects serve as critical pillars in establishing a robust foundation for supporting simultaneous interactions among multiple users.

The first aspect involves the synchronization of graphics across different devices. It is essential to achieve real-time synchronization of positions, interactions, and movements of individuals and holograms among all participants. This requires the creation of a shared virtual environment, often referred to as a "shared playground," where all the devices are interconnected and aware of shared objects. This synchronization challenge can be addressed using a client-server model, in which a central server manages the networking aspects to ensure consistent and simultaneous experiences for all users involved.
The second consideration involves aligning coordinates across multiple devices within a common local space, enabling holograms to appear in the same location for all participants. This becomes particularly important when using devices like the Hololens, which establish their starting position as the reference point for deploying the application. If two users do not start at the exact same position, although they may perceive the same virtual scene, the positions of the respective virtual objects will not correspond accurately. This misalignment occurs because each user's local coordinate system may differ, resulting in inconsistencies within the global coordinate system.

3.3 Research outcomes

Table 1: A table of the different solutions identified for implementing multi-user capabilities

Name	**Information**	**Appreciation**
Microsoft Shared experiences (Link [4])	Supports Microsoft Hololens 2; uses local rendering (1̃00k polygon limit) with Photon SDK for scene updates and Azure Anchors	(+) Has a tutorial; (+) compatible with Unity; (+) official solution from Microsoft; (-) uses the multiplayer framework Photon that is a non-Microsoft product; (-) uses Azure spatial anchors that are created via the Microsoft Azure: free account only valid for 1 year
NetCode for GameObjects networking library (Link [5])	Supports Microsoft Hololens 2; is a high-level networking library built for Unity; uses MLAPI (Mid-level API)	(+) Compatible with Unity
Mirror library (Link [6])	High level Networking library for Unity	(+) Compatible with Unity; (-) is not compatible with .NET Backend on UWP because of missing APIs. See documentation: Missing .NET APIs in Unity and UWP
ShARe (Link [7])	Supports Microsoft Hololens 2; an instrumented head-mounted display with a projector and a server motor attached, enabling to display virtual content of the HMD-user to the environment on any planar surface	(+) No need for a second Hololens to share the experience; (+) non HDM-user is able to interact with the virtual content; (-) custom design
Unreal Engine's built-in networking system (Link [8])	Supports Microsoft Hololens 2; uses a client/host architecture (client/listener); prerequisites : Unreal Engine 4.26+ and Microsoft OpenXR (API) plugin	(+) Has a tutorial; (-) solution does not use Unity
Haro3D library version 2 (Link [9])	Supports Microsoft Hololens 2; uses Labview as software	(-) Haro3D library only has a 30-day free trial; (-) solution doesn't use Unity
ArUco marker tracking (Link [10])	Supports Microsoft Hololens 2	(-) Physical marker

The research findings are summarized in Table 1, which presents an overview of the relevant elements for this project. The selection of these elements was guided by their compatibility with the Microsoft Hololens, a crucial consideration in the decision-making process. One option explored was the utilization of AruCo marker tracking, which enables multiple devices to synchronize. However, this solution was not chosen due to the desire to minimize the number of installations required, particularly when working with patients.
Another interesting approach was proposed by ShARe, involving modifying the Hololens to accommodate a mounted projector, allowing the other player to see virtual objects without wearing a Hololens. While this solution eliminates the need for a second device, it has limitations in terms

of interaction with virtual objects and necessitates additional hardware.
Two other solutions, Haro3D library and the Unreal Engine's built-in networking system, were considered but deemed unsuitable as they do not align with the project's collaborative work with researchers from the HumanIST group, who primarily use the Unity platform. Additionally, the NetCode for Gameobjects networking library and Mirror were found to support the device and platform requirements, but their implementation required substantial investment and lacked sufficient documentation compared to the chosen solution.
Ultimately, Microsoft's Shared Experiences was selected as the preferred solution for implementation due to its strong alignment with the project's requirements and objectives.

3.4 Final choice

Shared Experiences, supported by Microsoft HoloLens 2 and developed using Unity as the platform, applies the architecture presented in Section 3.2. It leverages the Mixed Reality Toolkit (MRTK) framework, which coincides with the toolset utilized by other members of the HumanIST group involved in related projects. This alignment ensures compatibility and facilitates collaboration among different projects within the group.
Moreover, the selection of Shared Experiences is reinforced by its status as the official solution provided by Microsoft. Given its role as the device developers, this indicates a higher likelihood of long-term support and ongoing advancements. However, it is important to note that the chosen solution is not ready to use. From the analysis it was deduced that significant adjustments to various sections of the tutorial are necessary.

4 Implementation

This chapter focuses on highlighting the key characteristics of the final implementation necessary to support multi-user capabilities. Given the nature of the implementation, certain aspects are discussed in detail, taking into account the requirements for creating an application suitable for Low Vision Rehabilitation (LVR). This includes considerations for robustness, privacy, and usage.
As discussed in Section 3.2, there are two fundamental aspects for achieving multi-user capabilities: graphics synchronisation and establishing a shared coordinate system. These two aspects will serve as the main sections of this chapter and be explored in detail.

4.1 Graphics synchronisation

The creation of a shared virtual environment is accomplished through the utilization of a client-server model shown in Figure 4. This architectural framework facilitate the communication between a provider of a resource, called server, and service requester called clients. In this specific case, the client-server architecture operates as follows: when a client moves or manipulates a virtual object, they send a notification or request to the server via the internet. The server then processes this information. Simultaneously, other clients connected to the server are notified whether any changes have occurred in the scene[11].

Multiple approaches can be employed to implement this client-server architecture. One option is to develop a custom server tailored to specific requirements. Alternatively, Microsoft recommends utilizing the Photon Engine in their shared experiences tutorial. The Photon Engine is a widely adopted game engine with a global user base exceeding 20 million individuals. The focus hereof is on multiplayer game development and offers a comprehensive suite of products, software, technologies, and networking components that enhance speed, performance, and the overall online gameplay experience. It is important to note that certain features of the Photon Engine may require payment, typically based on the number of concurrent users they can support.
Applications developed using the Photon suite are hosted on the Photon Enterprise Cloud, which accommodates over 800,000 applications. It is hosted by remote servers that are maintained by the photon team. It is distributed across major world regions to ensure minimal latency.

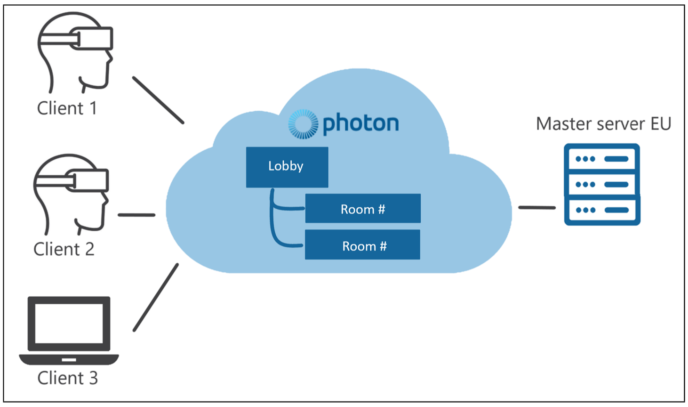


Figure 4: Overview of the client-server

As part of the Photon Engine ecosystem, the third-party provider offers the Photon Unity Networking (PUN) framework. PUN seamlessly integrates with the Unity framework, and its API closely resembles that of Unity. PUN simplifies the implementation of multiplayer games by providing the necessary functionalities and capabilities[12].

4.1.1 Photon Unity Networking

PUN is a Unity package designed specifically for developing multiplayer games. The package is structured with three layers of APIs to facilitate various functionalities. At the highest level, it provides Unity-specific features required for seamless integration. The second layer consists of the Realtime API, enabling communication and connection with the Photon servers. Lastly, the package incorporates DLL files encompassing serialization protocols for efficient data transfer.
PUN can be easily obtained from the Unity Asset Store. For this particular project, the free edition of PUN 2 (version 2.42) is utilized. Although it offers the same content as the paid version (PUN Plus), it has a limitation of hosting up to 20 concurrent users and is limited to 60 GB of traffic per month. Since this implementation is intended for LVR and from testing the solution, the provided user capacity is deemed sufficient as the traffic never exceeded 0.5 GB[13].

4.1.2 Implementation of PUN

PUN operates based on the following fundamental concepts. Initially, a photon application is created through the photon portal, which involves navigating to the Photon dashboard and registering a new account. The user then proceeds to create a PUN application as shown in the Figure 5, which will serve as the hosting environment for the shared virtual environment. After creating the application, an App ID is provided, which is used to setup the Unity project. This is done with the help of the PUN Setup Wizard shown in Figure 6 where the AppID has to be entered.
The sequence of the implementation in Unity is shown in the diagram presented in Figure 7. Initially, all users connect to a "Master Server" to initiate the process (1). The function that is called to establish the connection is the *PhotonNetwork.ConnectUsingSettings()*. It takes the entered AppID to ensure that the different applications are separated in the system and other various requirements for the connection.
After this stage, the user enters the Lobby (2). The Lobby, specific to the application, resides on the Master Server and functions to list available rooms for the game. The server then assigns the user to a particular room (3). The architecture of the Photon Cloud is designed for "room-based games". In the context of Photon, rooms establish communication boundaries, implying that users must be in the same room to share a virtual environment. Typically, in our application context (LVR), a room would consist in a rehabilitation session containing the therapist and the different participants. Several therapists could each run a concurrent rehabilitation session in their own room.
For the implementation at hand, a straightforward approach was adopted for room assignment

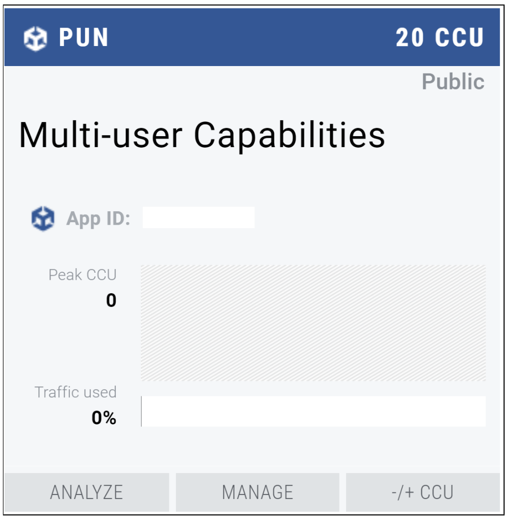


Figure 5: PUN application on the Photon dashboard

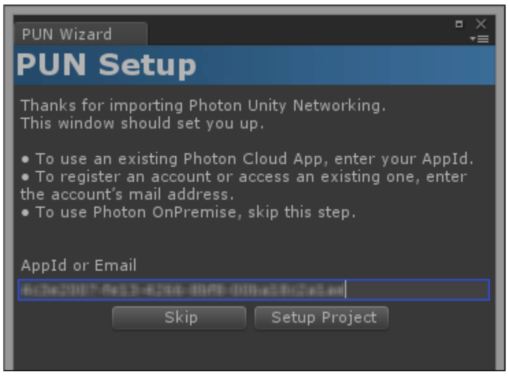


Figure 6: PUN setup for cloud registration

with the help of the function *PhotonNetwork.JoinRandomRoom()*. It works in this way: If a room already exists within the application, the Photon server redirects the user to that room with the help of the callback *OnJoinedRoom()*. In the absence of an existing room, a new room is created with the help of the callback *OnJoinRandomFailed()*. And if the user wants to quit the room and rejoin the lobby this can be easily done with the help of the function *OnCancelButtonClicked()* and then he can get attributed a new room. It is important to note that all these functionalities listed can be customized and adapted to meet specific requirements for our application context.
Afterwards, the player enters the room. Upon entry, they receive a list of players currently present in the same room, along with a nickname for identification purposes. The game then commences, and the various objects shared within the virtual environment are instantiated as networked GameObjects using the function *PhotonNetwork.Instantiate()* (4). This specific prefab takes responsibility for both writing and reading the networked object with a frequency of 30 hz [14], ensuring synchronized updates across all connected users.
For future reference, if new interactable objects are required, they can be directly added within the *CreateInteractableObjects()* function. And they need to have the the script *PhotonView* attached. Alternatively, in Unity, two serialized fields are set up to simplify the process of adding objects. Users have the option to select the desired user prefab and the prefab associated with the desired task.

4.2 Common coordinate system

There are several approaches to achieving a shared coordinate system in MR environments. One common practice is to utilize an anchor, which can be either physical or virtual. The purpose of the anchor is to serve as a common reference point among the devices, ensuring consistent positioning of networked objects for all users. The anchored object will be used as the root of the project and be shared between the devices. The other users will also set the anchored object at the root of their project. Consequently, by having each root at the same exact position you ensure that all other virtual objects that are children to this root will appear at the same position.
One option is to use physical anchors like ArUco markers, where users need to physically place objects in the real space as reference points. However, for our specific Low Vision Rehabilitation (LVR) project, implementing a virtual anchor proves more advantageous, as it eliminates the need for users to handle and position physical markers, thus enhancing usability.
By implementing a virtual anchor, the coordination of the shared coordinate system can be fully automated, requiring no intervention from the users and contributing to a more enjoyable user ex-

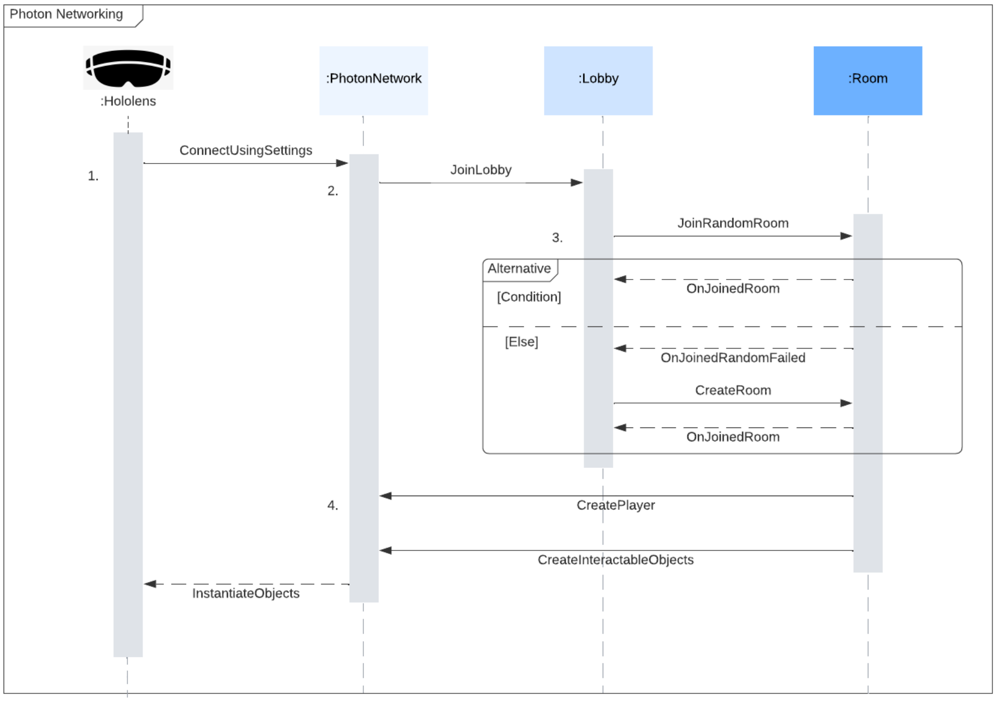


Figure 7: UML Sequence diagram of the Photon Network

perience. There are multiple methods available for creating a virtual fixed point in space through spatial anchor sharing. One option is to use a worldAnchor, which allows for local anchor sharing between devices. However, this approach may not provide the most robust solution as the anchor is shared by one Hololens to another directly. Therefore it needs to be stored on the individual devices and rely on a good connection between them.
Another commonly used method is employing Azure Spatial Anchors (ASA), a service provided by Microsoft. ASA enables the creation of durable cloud-backed spatial anchors that can be located across various devices, including HoloLens, iOS, and Android. Additionally, Microsoft has developed World Locking Tools that utilize a group of spatial anchors from the ASA service to lock the entire coordinate space relative to the physical world, rather than relying on individual anchors for specific objects, that has to be placed manually. World locking the entire space offers benefits such as improved precision in layout and greater efficiency in terms of both developer time and runtime resources[15]. However, we have determined that using an individual anchor from the ASA service is sufficient for our implementation. As our tasks are usually done in the same room and the movement is limited.

4.2.1 Azure Spatial Anchors

Azure Spatial Anchors (ASA) is a versatile developer service that facilitates the creation of mixed reality experiences. It enables the persistent positioning of virtual objects across multiple devices over a determined duration. This capability is achieved through the utilization of Mixed/Augmented Reality trackers, which leverage device cameras to sense and track the environment in six degrees of freedom (6DoF). This means that the user's head position, head movement, and overall orientation are taken into account[16].
ASA allow developers to designate specific points of interest in the real environment as "anchor" points. In our case, the anchor serves as the root of our shared coordinate system. When a spatial anchor is created, the client SDK captures environment information surrounding that point and transmits it to the ASA service. When another device searches for the anchor in the same physical space, similar data is transmitted and compared against the previously stored environment data.

If a match is found and sufficient environmental data has been recorded, the service sends the position of the anchor relative to the device's location.

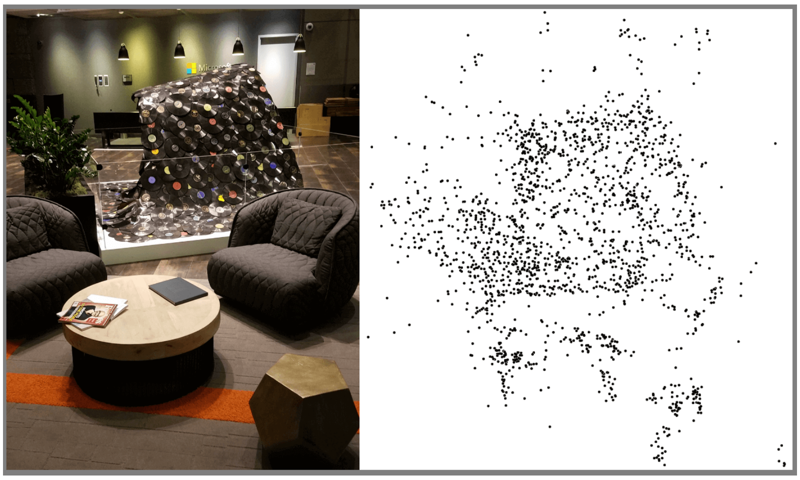

17

Figure 8: An environment and its derived sparse point cloud

The underlying process involves the device's camera capturing images of the environment, which are then processed into a derived format on the device. This derived format represents the environment as a cloud of sparse points as shown in Figure 8. Each point being associated with a hash that contains its characteristics. The derived format is then stored on the Azure cloud, requiring an internet connection. All stored data is encrypted using a Microsoft managed data encryption key, ensuring that anchors are isolated within the Azure account and can only be accessed by authorized applications. Additionally, the anchors are stored regionally based on the configuration of the application's Azure account, with available regions world-wide [18].
Moreover, the ASA cloud infrastructure implements a throttling limit. The different resource providers each have their own limit, for more information follow this link[19]. For the network throttling, the Microsoft Network provider applies a limit of writing/deleting (PUT) up to 1000 per 5 minutes and for reading (GET) up to 10000 per 5 minutes. For our testing and implementation purposes, this limit is more than sufficient. It is only for production deployment with high-scale requirements that it could become an issue[20].

When it comes to creating spatial anchors, the time required for the process is influenced by several factors, including network connection, device processing power and load, and the characteristics of the specific environment. It is crucial to emphasize the importance of placing anchors accurately, as a poorly positioned anchor can lead to an unreliable and ineffective user experience. Therefore, dedicating time to guide users in anchor placement is a critical design step that contributes to the overall robustness of the application. Certain guidelines should be followed to ensure successful anchor creation. Typically, in our application context, the anchor could be placed once by the therapist in the dedicated physical room and then reused each time the system is restarted so that the low-vision patient does not have to interact with it.
Firstly, stable visual features are essential for reliable anchor placement. This means selecting areas where the visual environment remains relatively unchanged. An example of a room with multiple discernible features vs a room with scarce discernible features is shown in Figure 9. Additionally, anchors should possess distinguishing characteristics, therefore it is not recommended to create anchors on large blank surfaces. Although anchors can float in the room without being physically placed on a surface, their positions should still be discernible. It is important to avoid highly reflective materials or rooms with many windows, as these factors can distort the understanding of the environment. Furthermore, repetitive patterns found on surfaces, such as carpets or wallpapers, should be avoided to ensure distinguishable characteristics for anchor creation. Lighting conditions should also be taken into consideration, as they can impact the detection of visual features by the application. Anchors created in strong natural light might be challenging to locate under artificial

lighting conditions, and vice versa.

[21]

Figure 9: An example of a good and bad room layout for ASA

Secondly, it is crucial to consider the user's viewing perspective and how they will attempt to locate the anchor later on. For instance, if virtual content is placed in the middle of a room, it is advisable to walk around it during the scanning process to ensure the anchor can be located from any position within the room. Scanning across the room captures a greater number of useful visual features compared to scanning a nearby wall. However, repeatedly moving the device from side to side while searching for an anchor is not helpful, as it only captures the same points from the same perspective. The key to achieving robustness lies in scanning the environment from the perspective of the individuals who will be trying to locate the anchor.
In certain situations, it can be beneficial to create multiple anchors. For instance, when dealing with significant lighting variations, having two anchors at the same location can offer a viable solution. Likewise, in an unstable environment, the presence of an additional anchor can help correct the position and improve stability. By strategically placing multiple anchors, the system can better adapt to changing conditions and enhance the overall performance of the application. This is the main idea behind the World Locking Tools, another product by Microsoft using a collection of anchors to map the space[22].
By adhering to these guidelines and considerations, the process of creating spatial anchors can be optimized, leading to a more robust and reliable multi-user experience. However, it is essential to recognize that evaluating the anchor's performance requires testing the application's anchor scenario in a real-world environment. In Chapter 5, we will implement an experimental design to thoroughly assess the implementation's limitations and performance in practical conditions. This empirical testing will provide valuable insights and validate the effectiveness of the spatial anchor implementation in real-life usage scenarios.

4.2.2 Implementation of Azure Spatial Anchors

To create a spatial anchor resource, the first step is to connect to the Azure portal. This portal requires a subscription to access various products and services. For the current work, a registered was made using a standard university account, which provided a free one-year subscription with certain limitations on storage and Azure Cosmos. However, for the current implementation, these capabilities were not utilized since the anchors were not stored long-term.
Once connected to the portal, a new resource of type "Spatial Anchors" needs to be created. This involves specifying several characteristics, including a unique resource name, a resource group for management purposes, and a location (region) to host the resource. Upon creation, the resource is assigned an identification, including an account ID, Account Domain, and Access Keys. Account Keys serve as credentials for application authentication with the Azure Spatial Anchors service, as illustrated in Figure 10. While in the development phase, using Account Keys is sufficient. However, as one progress towards production, it is recommended to transition to a more robust authentication mechanism, such as Access Tokens or Azure Active Directory user authentication. This allows for a better monitoring of the different users having access to the cloud. The main goal of all these elements are to configure our Unity project and to establish the connection between

the application and the resource.

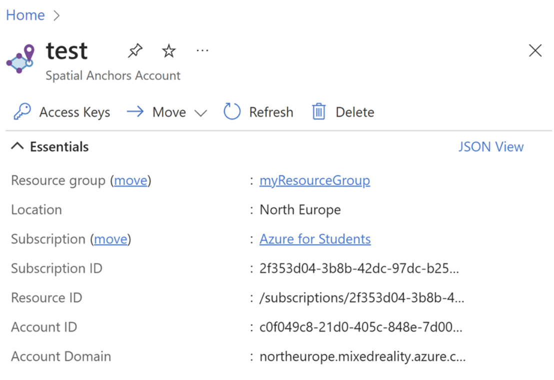


Figure 10: Screenshot from the Azure portal of a Spatial Anchor resource

The creation of spatial anchors is primarily handled by the *AnchorModuleScript.cs* file, which contains essential methods for this purpose. During application runtime, the user initiates the Azure session by invoking the *cloudManager.CreateSessionAsync()* method with the attributes that are retrieved from the Azure portal (1). This step establishes the connection with the Azure cloud. Subsequently, the creation of a spatial anchor begins (2). Initially, a new local cloud anchor is generated at the position of the anchor element. Its expiration date attribute, specifying the duration before automatic deletion from the cloud, is then set. For this implementation, the default 7-day limit is employed.

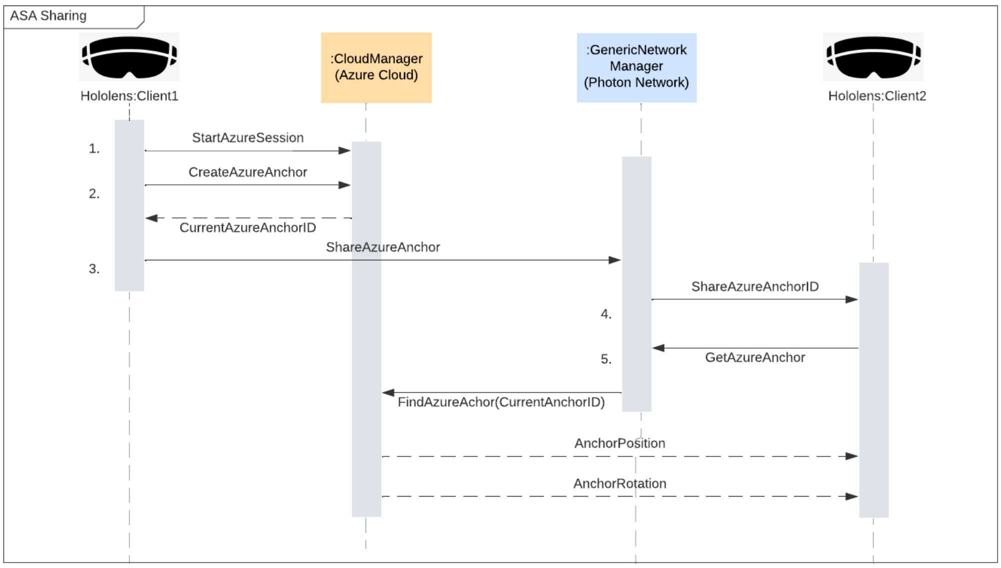


Figure 11: UML Sequence diagram of the sharing of ASA between users

As mentioned earlier, it is crucial to ensure sufficient capture of environment data before attempting to create a new cloud spatial anchor. This condition is verified, and if satisfied, the local anchor is saved by calling the *cloudManager.CreateAnchorAsync(localCloudAnchor)* method. In the implementation, this anchor is designated as the current cloud anchor and its identifier is shared with other users through the use of our PUN (Photon Unity Networking) implementation (3). The relevant methods for this functionality reside in the *SharingModuleScript.cs* file.
To facilitate synchronization among all players in the shared room, common instances are managed by the *GenericNetworkManager*. It maintains the current Anchor ID and updates it whenever a

new user shares a spatial anchor (4). The identifier is used to retrieve the anchor stored in the cloud by invoking the *FindAzureAnchor* method, with the current Anchor ID as a parameter (5). This is then sufficient for retrieving the data of the anchor and relocating our object with the anchored position. All these steps are shown in this sequence diagram in Figure 11.

4.3 Description of the global implementation

The final implementation incorporates the PUN implementation and the ASA, which were discussed in detail in previous sections. It can be found on this github repository[23]. For development purposes, the DebugWindow GameObject was created to display logs during various actions related to network connection and ASA creation. The Instruction GameObject was included to guide the user in creating a global spatial anchor. The user follows the instructions by interacting with the buttons provided within the ButtonParent GameObject. Additionally, there are components essential for the PUN implementation. These include the NetworkLobby and its child, the NetworkRoom. These components handle the necessary function calls described in the PUN implementation. The SharedPlayground GameObject encompasses all the virtual objects that are shared among users, including the ASA, which is a child of SharedPlayground. The ASA is named TableAnchor and serves as the root object where all other shared objects are placed. While the anchor can be made invisible, for clarity, it is represented as a blue cube positioned at the center of the scene. Each player has a white sphere attached to him, representing his virtual position. Moreover the shared object is a rover. It is a prefab that comes with the packages for enabling multi-user capabilities. The scene is shown in Figure 12 and a short demonstration can be found here[24].

Figure 12: Unity capture and real life picture of the implementation

5 Evaluation

The implementation was evaluated with the help of user feedback. The main goal was to evaluate the accuracy of the implementation and investigate how multiple users perceive the shared virtual environment by using the accuracy of shared virtual objects. For this evaluation a virtual red cube was shared between the participant and the examiner and its position was recorded with the help of a grid. The video of the evaluation can be viewed here[25].

5.1 Experimental design

A customized application was developed and implemented on the devices. This application comprised several components, including a virtual blue cube representing the Azure Spatial Anchor (ASA) and a virtual red cube with dimensions of 2.5 cm x 2.5 cm. The red cube served as a shared object between the participant and the examiner, allowing them to compare their respective positions. This setup not only enabled the assessment of object accuracy in the room but also facilitated evaluating the virtual positions of the players.

The advantage of using the red cube as a reference point was that if it appeared to be correctly placed, it indicated that all the shared objects in the room were also correctly positioned. This was possible because, thanks to the Photon Unity Networking (PUN), all the virtual elements were synchronized and placed in the same location relative to each other. Additionally, assessing the accuracy of a physically placed object was much simpler than measuring the virtual position of the player.
The participant and the examiner utilized different applications but were connected through the same PUN application and Azure Cloud, allowing for collaborative interaction.
The examiner's application included additional features, such as a debug screen that provided real-time feedback on every function deployed in the room. This feature allowed the examiner to monitor the application's performance and ensure its smooth operation. Furthermore, the examiner had specific buttons for creating and sharing the anchor, ensuring the synchronization process. In contrast, the participant's application had more limited functionalities. The participant could view the shared objects, including the blue and red cubes, but did not have the same level of control as the examiner. The participant's application provided buttons to initiate the Azure session and request the shared anchor, enabling them to interact with the environment and carry out the evaluation tasks.

5.2 Procedure

Eight participants were recruited (3 females and 5 males), seven without visual impairments and one with an eye condition known as strabismus. The age range of the participants was from 20 to 59 (mean= 27). Most participants had no prior experience with AR and VR.
The evaluation lasted approximately 45 minutes, with the actual wearing of the MR headset limited to around 25 minutes for safety and comfort reasons. The evaluation took place in a room measuring 8 m x 5 m, with a table in the center specifically designated for the experiment as shown in Figure 13. The blinds were fully closed and the two lamps were turned on. Red tape was placed on the floor to guide the participants and ensure accurate positioning.

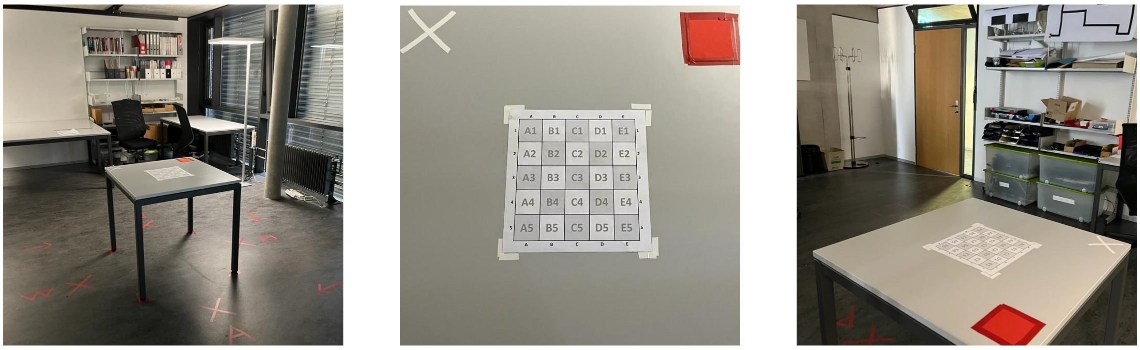


Figure 13: Evaluation room layout

The evaluation procedure, illustrated in Figure 14, consisted of three steps. In Step 1, two scenarios, namely LG (Large Grid) and SG (Small Grid), are presented to the participant. They are required to record the position of a red cube on the grid from four observation points: A, B, C, and D. In the first scenario LG, the cube is placed twice on a large grid, while in the second scenario SG, a small grid is used with two placements of the cube. Moving on to Step 2, participants are instructed to follow a predetermined path and move around accordingly. Finally, Step 3 is identical to Step 1, allowing us to examine the impact of movement on the recorded values.
The evaluation began by providing participants with instructions and an evaluation sheet, as shown in the appendix A. The sheet summarized the procedure, important considerations, and the consent given to participate. Participants were instructed to fill in the relevant sections during the simulation experience.
Next, the hardware setup for the MR system was introduced, including instructions on how to correctly wear the headset. The audio on the two devices were turned on and the luminosity was

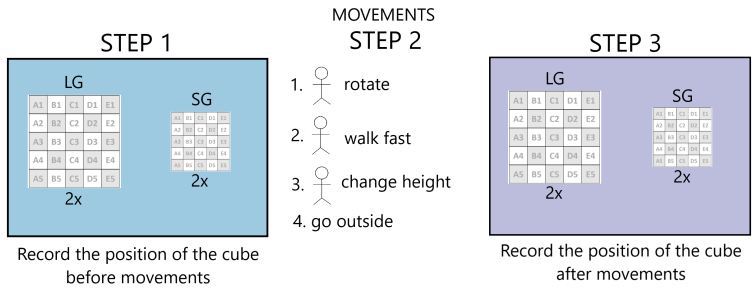
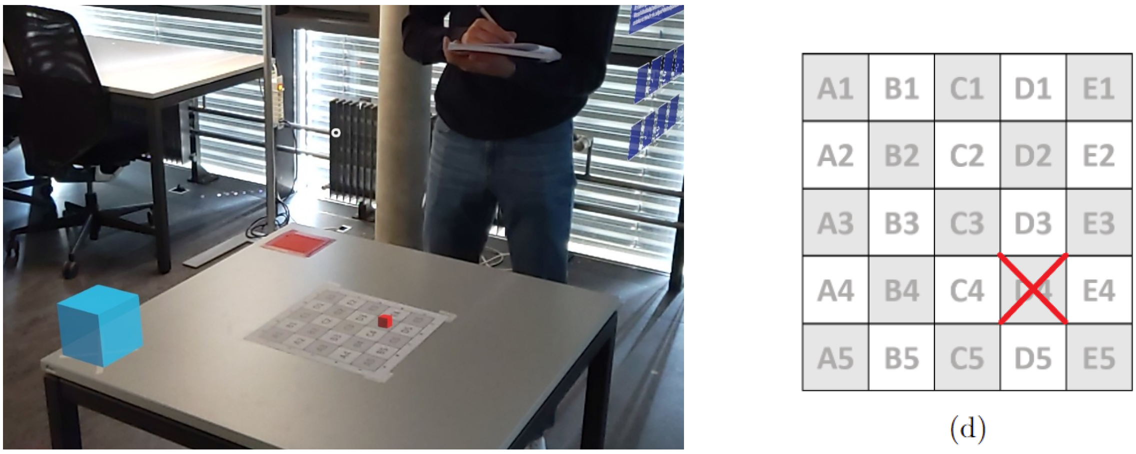


Figure 14: Evaluation procedure

at maximum level. The application was launched on both the examiner's and participant's devices. Participants were then guided to move around the room following the arrows on the floor. This allowed the devices to recognize the environment and create an accurate spatial representation. Once completed, the examiner manually positioned a blue cube on the table as the anchor and created an Azure Spatial Anchor (ASA) to share with the participant. The participant's application did not have the capability to create an ASA, so they were instructed to synchronize their device with the examiner's by pressing a button.
After successful synchronization, the step 1 of the evaluation procedure began. The examiner manually positioned the red cube on a 5 x 5 grid positioned on the table. To avoid bias, the participants turned their back and were instructed to look at a red cross on the wall. This was done so that the participants did not observe the examiner's movements. Once the cube was placed, the participant moved to four observation points marked by red tape around the table. They stood up straight and they marked the cube's position from the four designated points as shown in Figure 15. If the cube was between grid cells, the participant marked multiple cells on their sheet accordingly. This process was repeated four times: twice for scenario LG on a large grid with cell sizes of 4.5 cm x 4.5 cm and twice for scenario SG on a small grid with cell sizes of 3 cm x 3 cm. Prior to placing the cube at each designated position, the examiner would place it on a neutral position marked by a red cell on the table, as seen in Figure 13.



(d)

Figure 15: Participant recording the position of the red virtual cube

Following that, step 2 was conducted. The robustness of the implementation and its responsiveness to participant movements were assessed. Participants performed various movements, including rotation, changes in height, and fast movements. Additionally, they were asked to exit the room by opening and closing the door, moving 2 meters outside, and then reentering.
Finally, for the third step of the evaluation, the same procedure as for step 1 was repeated.

5.3 Results

The study identified limitations in accurately placing shared objects. To analyze the evaluation results, a script was used to visualize the data using the Seaborn library. Heatmaps were selected as the visualization method due to their ability to clearly represent the frequency of positions. Since four perspectives from the four observation points were recorded for each cube, heatmaps provided an easy visual indication of which positions were most commonly recognized. Four heatmaps following scenario LG and SG before and after movements were generated based on the results from each participant. The values within each cell of the heatmap represented the mean position recorded over a grid, considering the first and second cube placements as the cube was placed twice for each scenario. The middle cell of each grid (position (3,3)) was designated as the correct placement and highlighted with a green border for clarity.

The results of the evaluation are summarized by heatmaps in Figure 16. It represents the mean from the participants with no eye condition. It is important to note that the participant with strabismus was excluded from this particular analysis and will be discussed separately in its dedicated section 5.4 of the results. Moreover, since the participants were allowed to select multiple cells for the position of the cube, the correct cell could have a value of 100% as it was always selected. But still have positive values on the surrounding cells. Therefore for even more precision, the mean of the selected cells is not sufficient an the probability of the appearance of the center cell should also be calculated. This is done in Figure 20 discussed in chapter 6.

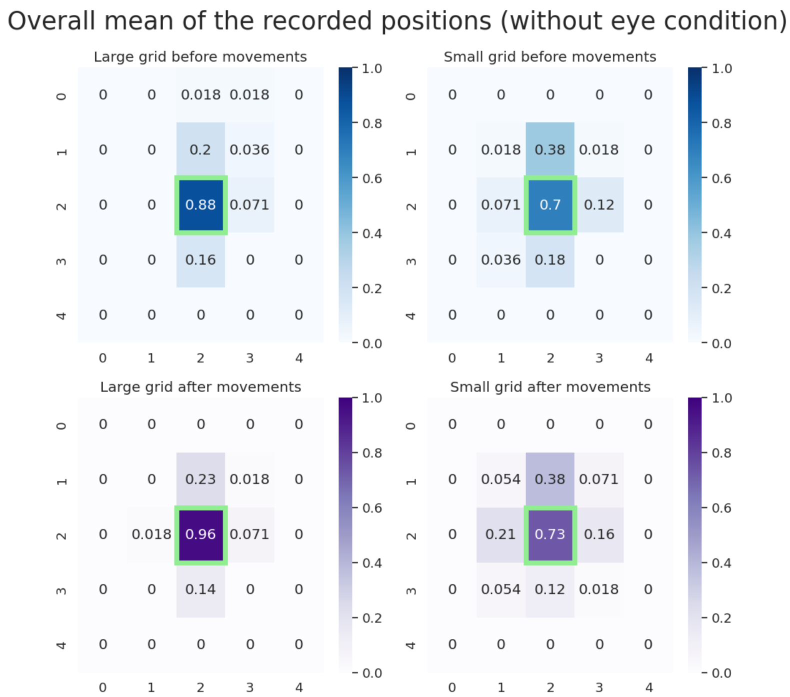


Figure 16: Mean of the results for every grid with regards to users with no visual condition

In the first step, scenario LG, participants were tasked with indicating the position of the red cube on a large grid. Out of the seven participants, three consistently identified the cube in the correct position for every viewing point while some still recording some of the surrounding cells. However, the remaining three participants (3, 5, 6 and 7) did not always perceive the cube in its correct position. Nonetheless, when considering the frequency of their viewing positions, these participants correctly identified the cube's position at least 75% of the time. During the evaluation, it became evident that the technology has certain limitations. The cube showed positional changes when viewed from different perspectives. As a result, all participants reported a small percentage of recorded positions in the surrounding cells. Notably, there was a tendency for these variations to be more prominent in the upper-right portion of the grid. Participant 6 consistently showed a

tendency towards the lower left position.
For the scenario SG, participants were asked to indicate the position of the red cube on a small grid. In this case, more variation was observed. Only two participants consistently located the red cube correctly. The other participants struggled to locate the cube, with participant 7 performing the worst, correctly identifying the position only 38% of the time. However, the remaining participants did better, recognizing the cube correctly more than half of the time. Similar to the first scenario, there was a frequency of small percentages in the upper-right portion of the grid, and participant 6 consistently showed a tendency towards the lower left position.
Next, for step 2, participants engaged in a movement exercise involving rotations, height differences, and exiting the room. After completing the exercise, new series of cubes were placed on the grids for step 3.
In the first scenario LG, five out of seven participants consistently saw the red cube in the correct position after the movements. This showed an improvement compared to the same scenario before the movements, and the result patterns closely resembled those observed previously.
In the scenario SG after the movements, the cube positions were also a bit better overall. However, in a few instances, there was a slight deterioration in performance. Additionally, the result patterns remained consistent with those observed in the second scenario before the movements.

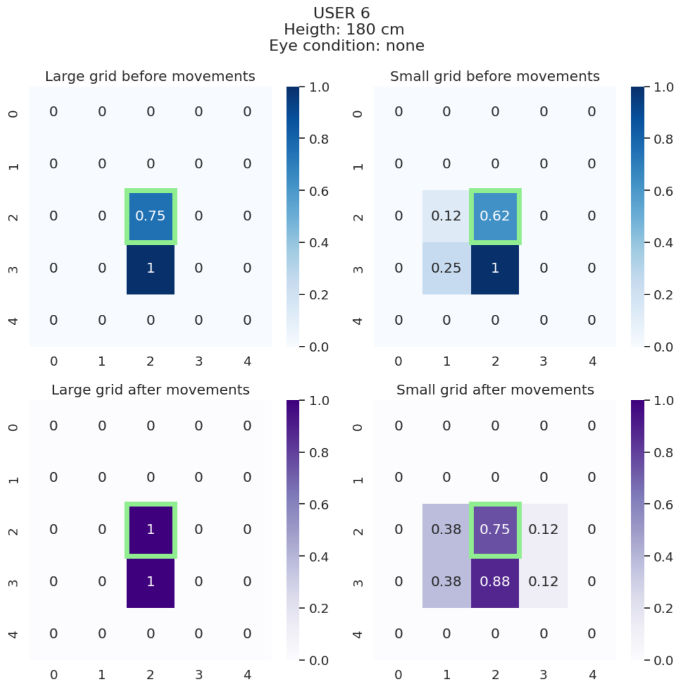


Figure 17: Heatmaps of the results from participant 6

The graphs from three participants in particular deserve attention. Participant 6 consistently had results that were shifted down by one cell, as depicted in Figure 17. This consistent shift suggests that the anchor did not initially locate correctly at the start of the exercise. As a result, both the cube and other elements in the room were shifted by a small amount. Since the evaluation relied on the accuracy of the initial anchored position, which was not refreshed throughout the exercise, this difference persisted.
The graphs from participants 5 and 7 from Figure 18 show a slight decrease in accuracy after the movements. However, the results are inconclusive. Interestingly, there is a slight improvement on the large grid but a decline on the small grid. This difference could be attributed to human error or the fact that the small grid magnifies any changes in the cube's placement, necessitating greater attention and consideration.
Overall, these observations highlight the importance of precise anchor placement and the potential impact of movements on the accuracy of the shared elements.

5.4 Additional results from a participant with strabismus (crossed eyes)

The evaluation also included a participant with a specific eye condition, characterized by left squinting. These results were intentionally excluded from the overall analysis to ensure the integrity of the findings. The eye condition affected the participant's depth perception, making it particularly challenging to discern the cube's position on the surface. The participant mentioned that at times, the shared object appeared more like a rectangle than a cube. Figure 19 displays the results from this evaluation.

The participant was able to locate the correct position almost every time, with an 88% accuracy on the small grid. What is intriguing is that not only was the correct position identified, but also the surrounding cells. This finding is more prominent compared to the results from other participants. It is likely attributed to the participant's crossed eyes, which typically result in diminished three-dimensional vision[26]. Nonetheless, it is impressive to observe the participant's consistent accuracy in determining the position despite this visual impairment.

Figure 18: Heatmaps of the results from participants 5 and 7

6 Discussion

The user study demonstrated that this implementation is a promising tool for multi-user capabilities. Participants generally showed the ability to accurately determine the correct position of the shared object within our study using the printed grids. In cases where the position was not immediately clear from one perspective, participants were able to move around and determine the most accurate position. As the heatmaps show the mean of all the times the different cells are selected it is also interesting to take into consideration the probability of only the correct cell being selected on the grid (For example: if the participant chose two cells then each cell would have a probabilities of 0.5). This results in the table shown in Figure 20. It has to be noted as this probability is a pessimist approach to our evaluation due to the fact that the results are binary, the participant crossed out the entire cell or not. Therefore the results did not distinguish cases where the cube was largely on the surrounding cells or only a small amount. Future evaluations could consider refining the data collection to provide even more precise results.

The probabilities give another perspective to the findings. They confirm the good results from the participants 1, 2, 4 and 5. The other participants that were identified as having problematic results are also highlighted. Interestingly the table shows that the step 3 has a lower probability of selecting the correct cell. So it would mean at this step that multiple cells were chosen for each step and therefore the probability of this single cell is smaller. This could maybe be due to the

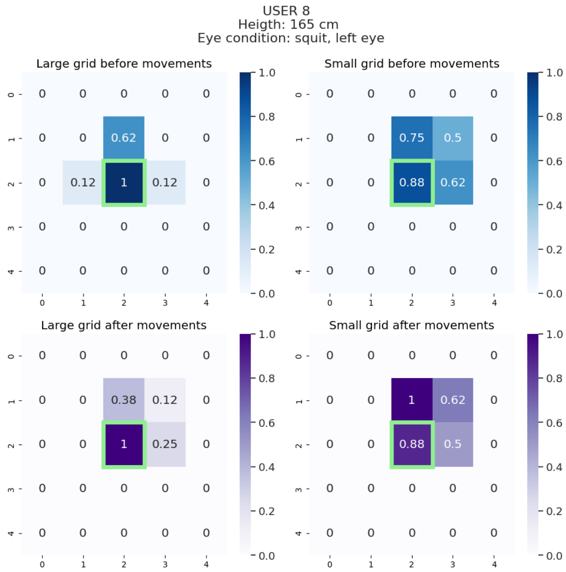


Figure 19: Heatmaps of the results from participants 8

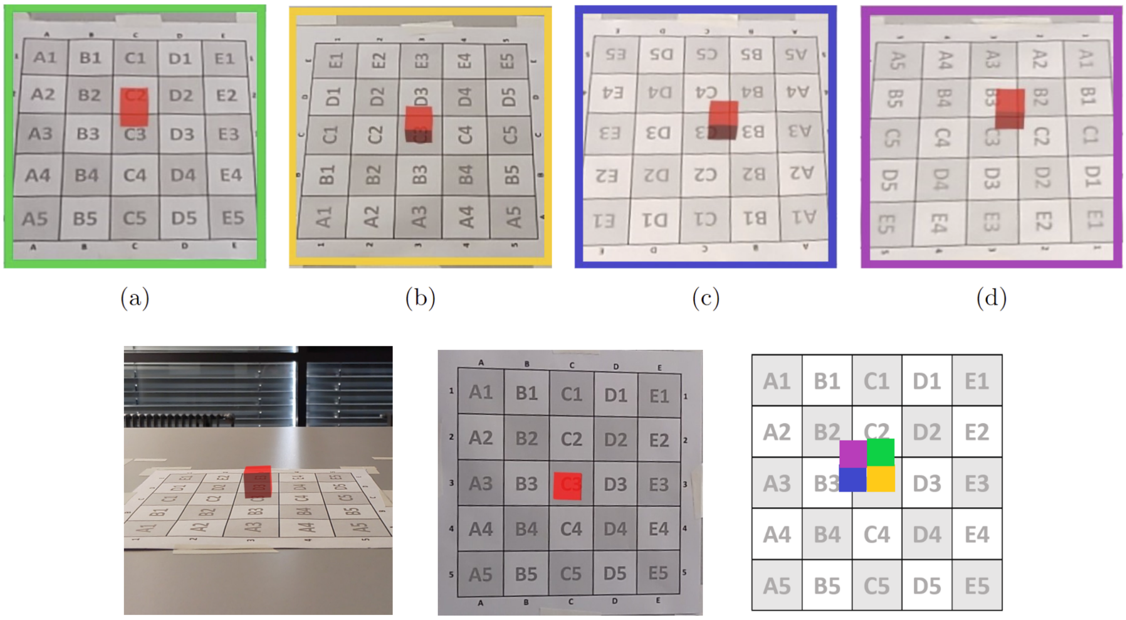
fact the the movements made the position of the cube less precise. Overall this table shows that there is are good accuracy for this implementation. And that is robust to movements as the values only decrease by a small amount.

	LG before movements	***SG before movements***	***LG after movements***	***SG after movements***
Participant 1	0.888889	0.466667	0.888889	0.4
Participant 2	0.727273	0.333333	0.8	0.538462
Participant 3	0.583333	0.8	0.8	0.727273
Participant 4	1	0.615385	0.8	0.571429
Participant 5	0.75	0.5	0.7	0.272727
Participant 6	0.428571	0.3125	0.5	0.285714
Participant 7	0.4	0.272727	0.4375	0.090909
All participants (with no eye condition)	0.636364	0.458824	0.666667	0.405941
Participant 8*	0.533333	0.318182	0.571429	0.291667

Figure 20: The probabilities of selecting the correct cell

It is worth noting that the accuracy of shared positions did not change users' ability to interact with the objects. Thanks to the Photon Unity Networking (PUN) system, the objects were updated simultaneously on all devices. Users were able to see to virtual objects and move them, just not always at the same global position. However, it can be peculiar for one user to see another user manipulating an object which appears to be not accurately grabbed from his perspective.
Participants provided valuable feedback regarding limitations of the HoloLens device during the study. Specifically, the device struggled to accurately represent objects with correct depth perception. As a result, the small red cube used in the study exhibited slight positional changes depending on the viewer's perspective. For instance, it could appear centered from one angle but off-center from another, as shown in Figure 21. A single Hololens device was used to capture the images of

the cube from different perspectives. The final image is a drawing of the different positions of the cubes. It highlight the importance of these position changes. This insight influenced the choice of grid sizes, with the larger grid designed to accommodate these differences, while the smaller grid aimed to showcase variations in greater detail. It is reasonable to assume that these depth representation limitations contributed to the observed variations in recorded positions among the study participants.



(a) (b) (c) (d)

Figure 21: Captures of the same red cube from different positions (markings a,b,c and d, horizontal, and directly over) on the small grid and a drawing of how the position changes over the perspective of the 4 markings

Additionally, the small size of the shared cube posed challenges in manipulation and movement. For example, when trying to grab it sometimes it did not register that the hand was near it and did not move. It is worth noting that this difficulty was more pronounced when using the older Hololens 2 device provided by the university. Consequently, the evaluation was conducted solely with the two most recent devices available. To address the issues related to object manipulation, it could be resolved with bigger bounding boxes around small virtual objects or using other modalities of interactions instead of direct grabbing. Additionally when designing virtual objects, larger sizes are preferable to maintain accuracy and depth perception, as larger objects exhibited less noticeable positional changes when viewed from different angles. Knowing these limitations, it is important to acknowledge that the examiner manually placed the cube, even if every viewing point was checked to ensure that the red cube was in the correct cell. Human error should be considered when interpreting the study results.

There was one evaluation session that could not be recorded due to technical difficulties with synchronizing the Azure Spatial Anchor (ASA). As a result, the shared virtual objects ended up with inconsistent positions. This highlights the significance of accurate ASA placement, as it heavily affects on the surrounding environment. In the current implementation, the ASA was only positioned once, which compromises the system's robustness. A potential solution could involve utilizing a group of ASAs that adjust their positions relative to one another, or employing World Locking Tools, which automatically identify key points in the room and update multiple anchors. These alternatives have the potential to address these limitations and improve the reliability of the system.

It is important to note that a reliable internet connection was essential for deploying the application. In the absence of an internet connection, the application had to be terminated and restarted. This was due to the fact that the shared cube was generated only when connected to the PUN

application, and without internet, the red cube would not be visible. Even if an internet connection was established later, the object generation only occurred during the initial initialization of the application and therefore could not be corrected afterwards. Furthermore, in cases where this issue occurred and another device joined the room, sometimes it would encounter the same problem, while other times it would not.
Finally, it was important to correctly terminate the application between sessions. This could be done by either accessing the settings on the HoloLens device to terminate the application or using the device portal for termination. Failure to terminate the application properly sometimes resulted in newly connected devices not correctly initialising the shared virtual objects.
The evaluation revealed several significant limitations but should also be considered critically. Firstly, it is important to note that the study was conducted on a small scale, involving only eight participants. In order to fully assess the functionality of the implementation, further testing with individuals who have low vision is necessary. Additionally, the evaluation encompassed a wide range of movements, making it challenging to precisely determine the impact of each movement on the final results.

Shifting our focus to the implementation, it has been shown, thanks to this evaluation, that it is a viable candidate for future collaborative tasks within MR. However, there are several points that need to be considered for future improvements.
Firstly, the implementation of the Azure Spatial Anchors was the most difficult part and took the longest. It has to be noted that it can only be used on a Hololens version 1 and 2 or an emulator, and it did not work just when it was run in Unity. Therefore, for testing the different components of the application, it had to be deployed each time on two different devices. Moreover, the current implementation does not utilize all the capabilities of the ASA. Only one anchor is used and stored in a variable. Each time a new anchor is created, the previous one is discarded and the new anchor is assigned a random ID and set as the new value of the variable. A more effective approach would be to assign specific IDs to anchors and retrieve them based on their assigned IDs. This would allow for better utilization of different anchors based on the specific task at hand. Additionally, it is not clear how long one can make the anchor persist in the Azure Cloud. Currently, it is set to 7 days, but this duration can be extended. One potential solution is to utilize Azure Cosmos DB [27], a database that supports long-term storage.
Furthermore, the current implementation requires manual placement of the ASA. While certain functionalities can be automated, such as user-controlled function calls that are implemented through buttons, it is not the most optimal solution for LVR. It would require the therapist conducting the session to have a deep understanding of the setup. A better approach would be to make it all automatic. This would require to implement World Locking Tools (WLT), which would eliminate the need for manual setup. From the documentation, WLT should be able to work on top of the current application, replacing the ASA. An example of such an implementation can be found here[28].
Additionally, the implementation relies on different servers to provide the necessary services. The use of the Azure Cloud, although currently on a free subscription, may not be ideal for long-term LVR tasks. It would require therapists to have their own subscriptions, resulting in additional fees. One potential solution is for the HumanIST group to connect their own Azure cloud by having a subscription and share an access token while monitoring access. However, this solution is not ideal, and further investigation is needed to determine if the ASA format can be used without requiring a subscription.
In terms of the PUN server, it is suitable for the context of LVR as the free version offers sufficient capacity for use cases, even with multiple rooms, as it accommodates up to 20 concurrent users. Scaling up can be achieved by using multiple PUN applications, as the Photon dashboard can host multiple applications simultaneously. However, the way rooms are assigned within the implementation needs improvement. Currently, users enter rooms with active participants. It is worth noting that during testing, exiting and reentering the application multiple times with the HoloLens resulted in incorrect room assignments. This limitation highlights the need for a more robust solution, such as controlling the assignment of rooms by having distinguishable room names and allowing users to enter the desired room without relying on the presence of other devices. This

can also be solved as PUN already provides the necessary functions to implement this.

In conclusion, the findings from this research shed light on the implementation and limitations of the collaborative task in LVR. The study results highlight the potential of the system for multi-user interaction, while also revealing areas that require further improvement.

7 Conclusion

In this study, we explored the concepts of multi-user capabilities using the Microsoft Hololens 2 devices, aiming to address the lack of collaboration in MR, specially in regards to innovative low vision rehabilitation (LVR). Our exploration involved a comprehensive analysis of technical solutions for implementing collaborative features. After examining various options, we determined that using Microsoft Shared Experiences was the most suitable solution, as it addressed the necessary features of graphic synchronization and a common coordinate system.
We proceeded to implement a proof-of-concept using the selected solution, incorporating the PUN server for graphics synchronization and an azure spatial anchors (ASA) for achieving the common coordinate system. Each feature was presented in detail, including the employed technology, its limitations, and its usability.
To evaluate the implementation's limitations and feasibility, we conducted an experimental design focused on examining the position of a shared virtual object. The user study involved seven participants without eye conditions and one participant with strabismus. The results demonstrated that the position of the shared virtual object could be mostly recognized correctly. However, it also highlighted limitations related to the need for a robust and accurate shared coordinate system, as well as challenges in displaying virtual objects with correct depths using MR technology, leading to variations in positions from different viewer perspectives. It is important to note that this evaluation was conducted on a small scale and should be considered as an initial step for further research and technology evaluation.
This research fills a significant gap in the literature by addressing and empirically measuring the limitations and feasibility of a collaborative solution in MR, with additional considerations for LVR. The study contributes to advancing our understanding of collaborative aspects and lays the groundwork for future developments. Researchers in the field of mixed reality, such as the Human-IST group, can benefit from these findings as they open up opportunities for more immersive and interactive multi-user experiences.
In conclusion, this study contributes to the field of LVR by enhancing our understanding of collaborative aspects and paving the way for future advancements. The findings emphasize the importance of addressing the limitations and feasibility of collaborative solutions and provide valuable insights for researchers working in MR.

References

[1] Suyang Dong, Amir H. Behzadan, Feng Chen, and Vineet R. Kamat. Collaborative visualization of engineering processes using tabletop augmented reality. *Advances in Engineering Software*, 55:45–55, 2013.

[2] Duy-Nguyen Ta Huynh, Karthik Raveendran, Yan Xu, Kimberly Spreen, and Blair MacIntyre. Art of defense: a collaborative handheld augmented reality board game. In *Proceedings of the 2009 ACM SIGGRAPH symposium on video games*, pages 135–142, 2009.

[3] S Karthika, P Praveena, and M GokilaMani. Hololens. *International Journal of Computer Science and Mobile Computing*, 6(2):41–50, 2017.

[4] Stephan Lukosch, Mark Billinghurst, Leila Alem, and Kiyoshi Kiyokawa. Collaboration in augmented reality. *Computer Supported Cooperative Work (CSCW)*, 24:515–525, 2015.

[5] Yolanda Matas, Felix Hernandez-Del-Olmo, Elena Gaudioso, and Carlos Santos. An adaptive, comprehensive application to support home-based visual training for children with low vision. *IEEE Access*, 7:169018–169028, 2019.

[6] Susanna Nilsson, Bjorn Johansson, and Arne Jonsson. Using ar to support cross-organisational collaboration in dynamic tasks. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12. IEEE, 2009.

[7] Sebeom Park, Shokhrukh Bokijonov, and Yosoon Choi. Review of microsoft hololens applications over the past five years. *Applied sciences*, 11(16):7259, 2021.

[8] Georg Regal, Elke Mattheiss, David Sellitsch, and Manfred Tscheligi. Mobile location-based games to support orientation & mobility training for visually impaired students. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, New York, NY, USA, 2018. Association for Computing Machinery.

[9] Jun Rekimoto. Transvision: A hand-held augmented reality system for collaborative design. In *Proceeding of Virtual Systems and Multimedia*, volume 96, pages 18–20, 1996.

[10] Lauren Thevin, Carine Briant, and Anke M Brock. X-road: virtual reality glasses for orientation and mobility training of people with visual impairments. *ACM Transactions on Accessible Computing (TACCESS)*, 13(2):1–47, 2020.

[11] Wei Wang, Xingxing Wu, Guanchen Chen, and Zeqiang Chen. Holo3dgis: Leveraging microsoft hololens in 3d geographic information. *ISPRS International Journal of Geo-Information*, 7(2), 2018.

[12] World Health Organization (WHO). *World report on vision*, volume 214. 2019.

[13] Yuhang Zhao, Cynthia L Bennett, Hrvoje Benko, Edward Cutrell, Christian Holz, Meredith Ringel Morris, and Mike Sinclair. Enabling people with visual impairments to navigate virtual reality with a haptic and auditory cane simulation. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–14, 2018.

[14] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE, 2008.

Notes

[1]https://www.salus.edu/Colleges/Education-Rehabilitation/Low-Vision-Rehabilitation.aspx, consulted 23/03/2023, https://www.frontiersin.org/articles/10.3389/fnbeh.2015.00192/full consutled 23/03/2023, https://www.ign.com/games/hololens-young-conker, consulted 23/03/2023

[2]https://www.frontiersin.org/articles/10.3389/frvir.2021.647997/full, consulted 22/06/2023

[3]https://www.fireware.nl/en/optie/microsoft-hololens-2-industrial-edition-en/, consulted 22/06/2023

[4]https://learn.microsoft.com/en-us/windows/mixed-reality/design/shared-experiences-in-mixed-reality, consulted 22/06/2023

[5]https://docs-multiplayer.unity3d.com/netcode/current/about/index.htmlhttps://docs-multiplayer.unity3d.com/netcode/current/about/index.html, consulted 22/06/2023

[6]https://mirror-networking.com/, consulted 22/06/2023

[7]CMSY10(uni-ulm.de), consulted 22/06/2023

[8]https://github.com/microsoft/mixedreality-unreal-samples/tree/master/HL2Collab, consulted 22/06/2023

[9]https://forums.ni.com/t5/3D-Vision/Sharing-Holograms-between-Multiple-Hololens-using-LabVIEW/td-p/3687489, consulted 22/06/2023

[10]https://github.com/doughtmw/ArUcoDetectionHoloLens-Unity, consulted 22/06/2023

[11]https://www.geeksforgeeks.org/client-server-model/, consulted 22/06/2023

[12]https://www.photonengine.com/pun, consulted 22/06/2023

[13]https://doc.photonengine.com/pun/v2/getting-started/pun-intro, consulted 22/06/2023

[14]https://docs-multiplayer.unity3d.com/netcode/current/learn/clientside_interpolation/, consulted the 08/06/2023.

[15]https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/shared-experiences-in-unity, consulted 22/06/2023

[16]https://learn.microsoft.com/en-us/azure/spatial-anchors/overview, consulted 22/06/2023

[17]https://learn.microsoft.com/en-us/azure/spatial-anchors/spatial-anchor-faq, consulted 22/06/2023

[18]https://learn.microsoft.com/en-us/azure/spatial-anchors/spatial-anchor-faq, consulted 22/06/2023

[19]https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/request-limits-and-throttling, consulted 22/06/2023

[20]https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/request-limits-and-throttling, consulted 22/06/2023

[21]https://learn.microsoft.com/en-us/azure/spatial-anchors/concepts/guidelines-effective-anchor-experiences, consulted 22/06/2023

[22]https://learn.microsoft.com/en-us/azure/spatial-anchors/concepts/guidelines-effective-anchor-experiences, consulted 22/06/2023

[23]https://github.com/mikkeline-elleby/Bachelor-Project-collaborative-MR-AR-with-PUN-and-ASA

[24]https://www.youtube.com/watch?v=YWDDEXeGmyg&t=24s

[25]https://www.youtube.com/watch?v=55lS9Eusrwg&t=122s

[26]https://my.clevelandclinic.org/health/diseases/15065-strabismus-crossed-eyes

[27]https://learn.microsoft.com/en-us/azure/cosmos-db/introduction, consulted 10/06/2023

[28]https://github.com/jdcast/ar-final-project, consulted 22/06/2023

Appendices

A Instruction and evalution sheet for the evaluation

HUMAN-IST
HUMAN CENTERED INTERACTION SCIENCE & TECHNOLOGY

INSTRUCTION AND EVALUATION

June 22, 2023

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Objectives: This exercise aims at evaluating the feasibility's and limitations of the multi-user capabilities using the Microsoft Hololens 2[1].

Procedure: You will be wearing the device continuously throughout the evaluation. For initialization purposes you will need to interact with a button to synchronise your device with the examiners. Then you will be presented two grids of different sizes. The examiner will place a red cube in the grid and it is required to indicate the location of each cube by noting it on the provided sheet. This task will be performed 6 times. Afterward, you will engage in movement exercises, and afterwards be shown a series of cubes as before writing down their positions.

Risks: The use of augmented reality devices can cause nausea, dizziness, or sweating in some users, especially during prolonged use. To minimize these risks, we have limited the time to approximately 25 minutes. However, if you experience any of these symptoms, please inform the experimenter immediately. Additionally, if you have epilepsy, you must immediately withdraw from the study as the use of augmented reality could trigger seizures.

Contact: If you have any further questions about the study, the procedure, or the results, please contact Mikkeline Elleby (mikkeline.elleby@unifr.ch).

Consent: The objective and nature of this research have been sufficiently explained above, and I agree to participate in this study. Furthermore, I give consent for the aforementioned data to be recorded throughout the experiment, and I understand that no information about my identity will be included in the data. Lastly, I understand that I am free to withdraw from the study at any time and can contact the researchers with any questions or concerns I may have regarding this study.

Time Required: The entire study is expected to take approximately 45 minutes.

- ∼10 minutes for the introduction and explanations
- ∼25 minutes for the experiment
- ∼10 minutes for the conclusion

Instructions: For each step of the evaluation, a set of instructions will be given. They are written as a heading. Please read them all before beginning the examination and ask if there is any questions.

[1] https://www.microsoft.com/en-us/hololens

Before the examination

1. **Do I have any eye conditions:** ______________________________
2. **What is my height:** ______________________________
3. **I have experience with virtual reality:**
   ☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree
4. **I have experience with augmented reality:**
   ☐ Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Initialisation step

To begin the collaborative experience please press when asked to, on two buttons: start Azure session and get shared anchor These two steps need to be performed in order to synchronize your device with the examiner's.

Step 1: Objects position

For this first part of the exercise you will have to turn so that you are not facing the table and the examiner will place a red virtual cube on two different sized grid. You will have to check the different fields on the paper where you see that the red cubes is placed. For each cube you will need to write it down as seen from 4 different positions marked with red crosses on the floor.

LARGE GRID: first placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

LARGE GRID: second placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

SMALL GRID: first placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

SMALL GRID: second placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

Step 2: Move around

For this step we will need to move around and afterwards we will do the grid exercise once more. Do the following exercises, the examiner will guide you:

- Turn around 4 times on the markings on the floor
- Go around the room once in a fast pace
- Go one your knees then on your toes (repeat 3 times)
- Exit the room (until the red line) and reenter (about 2m to the right)

Step 3: Objects position after movements

For last first part of the exercise you will have to turn so that you are not facing the table and the examiner will place a red virtual cube on two different sized grid. You will have to check the different fields on the paper where you see that the red cubes is placed. For each cube you will need to write it down as seen from 4 different positions marked with red crosses on the floor.

LARGE GRID: first placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

LARGE GRID: second placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

SMALL GRID: first placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

SMALL GRID: second placement

Now go from step a to d, crossing out each times where the cube is seen on the grid.

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(a)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(b)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(c)

A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4
A5	B5	C5	D5	E5

(d)

When finished turn around and wait for the examiner to place the red cube and turn around when told to.

B Versions

Versions installed in the current Unity repository:

Azure Spatial Anchors SDK Core Version 2.13.3 `com.microsoft.azure.spatial-anchors-sdk.core`

Azure Spatial Anchors SDK for Windows Version 2.13.3 `com.microsoft.azure.spatial-anchors-sdk.windows`

Mixed Reality OpenXR Plugin Version 1.8.0 `com.microsoft.mixedreality.openxr`

Mixed Reality Toolkit Foundation Version 2.8.3 `com.microsoft.mixedreality.toolkit.foundation`

Mixed Reality Toolkit Standard Assets Version 2.8.3 `com.microsoft.mixedreality.toolkit.standardassets`

ARCore XR Plugin Version 4.1.7 - April 08, 2021 `com.unity.xr.arcore`

ARKit XR Plugin Version 4.1.7 - April 08, 2021 `com.unity.xr.arkit`

JetBrains Rider Editor Version 3.0.16 - October 12, 2022 `com.unity.ide.rider`

OpenXR Plugin Version 1.7.0 - March 07, 2023 `com.unity.xr.openxr`

Test Framework Version 1.1.33 - July 19, 2022 `com.unity.test-framework`

TextMeshPro Version 3.0.6 - April 22, 2021 `com.unity.textmeshpro`

Timeline Version 1.4.8 - May 04, 2021 `com.unity.timeline`

Unity UI Version 1.0.0 - November 08, 2022 `com.unity.ugui`

Version Control Version 1.17.6 - October 10, 2022 `com.unity.collab-proxy`

Visual Studio Code Editor Version 1.2.5 - February 09, 2022 `com.unity.ide.vscode`

Visual Studio Editor Version 2.0.16 - June 21, 2022 `com.unity.ide.visualstudio`

C Video for the implementation

`https://www.photonview.com/watch?v=YWDDEXeGmyg`

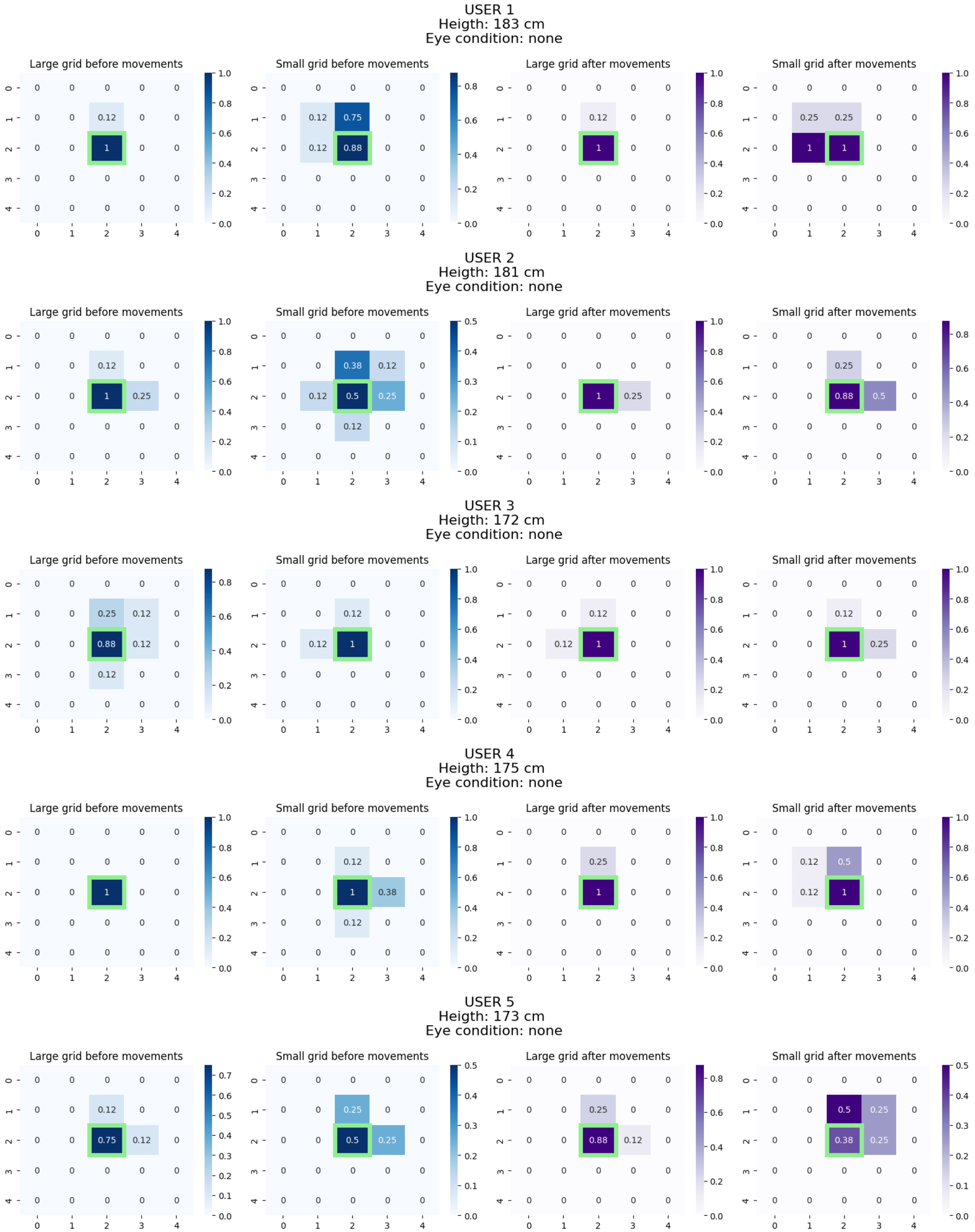
D Video for the evaluation procedure

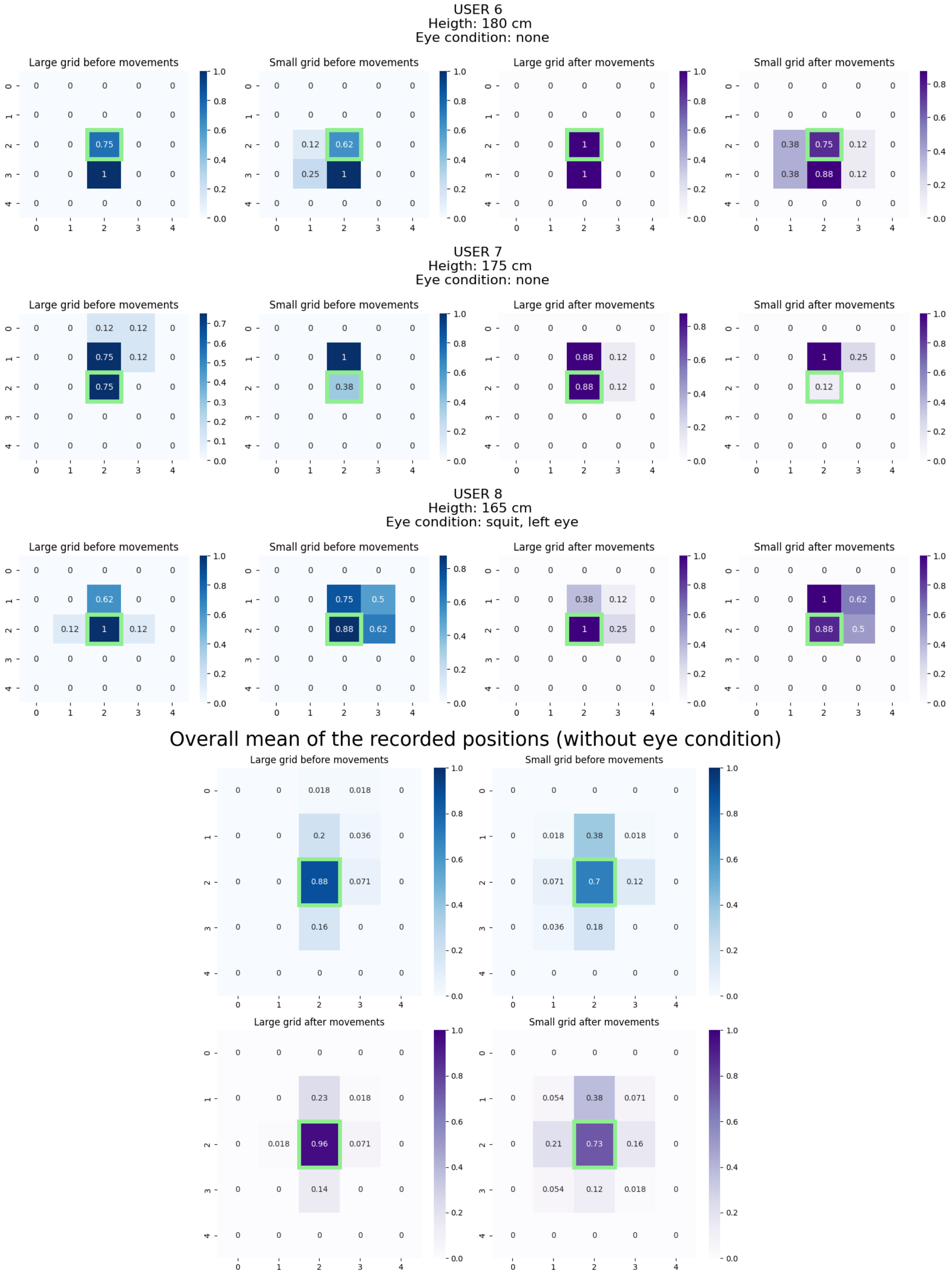
`https://www.youtube.com/watch?v=55lS9Eusrwg&t=122s`

E Github repository

`https://github.com/mikkeline-elleby/Bachelor-Project-collaborative-MR-AR-with-PUN-and-ASA`

F Results from the evaluation


USER 1
Heigth: 183 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 2
Heigth: 181 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 3
Heigth: 172 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 4
Heigth: 175 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 5
Heigth: 173 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements


USER 6
Heigth: 180 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 7
Heigth: 175 cm
Eye condition: none
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
USER 8
Heigth: 165 cm
Eye condition: squit, left eye
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements
Overall mean of the recorded positions (without eye condition)
Large grid before movements
Small grid before movements
Large grid after movements
Small grid after movements