Universidade de Aveiro
2023

Ivo Miguel dos Santos Félix

Framework de Assistência Remota para Manutenção Através de Realidade Virtual e Aumentada

Remote Assistance Framework for Maintenance Through Virtual & Augmented Reality

Universidade de Aveiro
2023

Ivo Miguel dos Santos Félix

Framework de Assistência Remota para Manutenção Através de Realidade Virtual e Aumentada

Remote Assistance Framework for Maintenance Through Virtual & Augmented Reality

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática , realizada sob a orientação científica do Doutor Paulo Dias, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Bernardo Marques, Investigador Auxiliar do Instituto de Engenharia Eletrónica e Informática de Aveiro da Universidade de Aveiro.

o júri / the jury

presidente / president	Professor Doutor Tomás António Mendes Oliveira e Silva Professor Associado, Universidade de Aveiro
vogais / examiners committee	Professor Doutor Daniel Filipe Martins Tavares Mendes Professor Auxiliar, Universidade do Porto - Faculdade de Engenharia
	Professor Doutor Paulo Miguel de Jesus Dias Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar gostaria de agradecer ao meu orientador, Paulo Dias, e co-orientador, Bernardo Marques, pela inexcedível disponibilidade e acompanhamento prestado. Aos meus colegas Rafael Fernandes e Gonçalo Junqueira, pelo apoio e momentos de diversão, com quem partilhei a jornada deste curso e que com orgulho iremos agora terminar, esperemos que com sucesso. Ao meu colega Álvaro Freixo, outro companheiro de caminhada nesta dissertação, tanto pelo suporte moral como também pela assistência em algumas tarefas deste trabalho, tais como o acompanhamento dos participantes e recolha de dados durante o estudo de utilizadores. Ao Tiago Davi, também pelo apoio, calma e ponderação que sempre transmitiu. E por fim aos meus pais, pelo apoio e por tornarem possível esta experiência neste curso, que sinto que verdadeiramente transformou a minha vida.

Palavras Chave

Assistência Remota, Realidade Aumentada, Realidade Virtual, Realidade Mista, Trabalho Cooperativo Apoiado por Computador, Réplica Virtual.

Resumo

A crescente especialização de profissões e globalização económica frequentemente produzem situações para as quais um especialista é necessário no local, mas a co-localização não é possível ou conveniente devido a considerações de tempo e custo. Ligar remotamente o especialista com um técnico no local é uma solução viável, mas a assistência remota para tarefas de manutenção apresenta vários desafios que não são facilmente satisfeitos pelas abordagens tradicionais de videoconferência, já que a informação espacial não é facilmente expressa desse modo. Tecnologias de realidade virtual e aumentada, por outro lado, prestam-se bem a este desígnio, visto que os objetos virtuais são referenciados espacialmente. Tomando partido destas vantagens, numa tentativa de responder aos desafios da proliferação do trabalho remoto, um *framework* de assistência remota para tarefas de manutenção usando realidade virtual e aumentada foi desenvolvido, com suporte para: assistência assíncrona; independência de vistas dos utilizadores através de reconstrução e modelação 3D do ambiente físico do técnico no local; comunicação através de pistas não-verbais através de anotações; réplicas virtuais como anotações. A combinação destas características é incomum na investigação atual e possibilita um alto grau de autonomia entre os utilizadores, com potencial para um uso mais eficiente do tempo do especialista remoto. Findo o desenvolvimento, o sistema foi avaliado com um estudo de utilizadores aplicado ao caso de estudo de manutenção de um edifício. O sistema obteve melhores resultados que os métodos tradicionais para a taxa de falha de tarefas, enquanto o tempo de execução foi melhor apenas para tarefas que não requeriam inserção de texto. Ademais, mostrou promessa como ferramenta de treino, e foi bem recebido pelos utilizadores.

Keywords

Remote Assistance, Augmented Reality, Virtual Reality, Mixed Reality, Computer Supported Cooperative Work, Virtual Replica.

Abstract

The increasing job specialization and globalized economy often produce situations for which an expert is required on-site but co-location is either not possible or convenient due to time or cost constraints. Connecting the expert with a local technician remotely is a viable option but remote assistance for maintenance tasks presents several challenges that are not satisfied efficiently by traditional video-conference approaches, as spatial information is not easily conveyed through these mediums. Virtual and augmented reality technologies, on the other hand, are well suited to that task as the information added as virtual objects is spatially referenced. Making use of these advantages, in an attempt to address the challenges of the proliferation of remote work, a remote assistance framework for maintenance tasks making use of virtual and augmented reality was developed, with support for: asynchronous assistance; independent user views through 3D model reconstruction of the local technician's environment; communication through non-verbal cues using annotations; virtual replica annotations. The combination of these characteristics is uncommon in current research and provides a high degree of autonomy between users, with the potential for a more efficient use of the remote expert's time. After development, the system was evaluated with a user study targeting a building maintenance use case. The system performed better than traditional methods in task failure rate, while completion time was better only for the tasks that did not require text-insertion. Furthermore, it showed promise as a training tool, and garnered positive feedback from users.

Contents

List of Figures

List of Tables

Glossary

6DOF	Six Degrees of Freedom
AR	Augmented Reality
AV	Augmented Virtuality
CMMS	Computerized Maintenance Management System
CSCW	Computer Supported Cooperative Work
FOV	Field of View
HCD	Human-Centered Design
HMD	Head-Mounted Display
HVAC	Heating, Ventilation and Air Conditioning
IVR	Immersive Virtual Reality
KPI	Key Performance Indicators
MRTK	Mixed Reality Toolkit
MR	Mixed Reality
NA	Not Available
OK/NOK	Okay/Not-Okay
QR	Quick Response
RGB-D	Reg Green Blue Depth
SAR	Spatial Augmented Reality
SDK	Software Development Kit
SUS	System Usability Scale
UI	User Interface
UX	User Experience
VE	Virtual Environment
VR	Virtual Reality
WLT	World Locking Tools
WO	Work Order
XR	eXtended Reality

CHAPTER 1

Introduction

On this chapter, the motivation, objectives and contributions of the dissertation will be presented.

1.1 Motivation

In today's globalized economy it is not always possible to have an expert on-site when required, and these situations are only expected to become increasingly common, given that as mankind's pool of collective knowledge increases so does the specialization of professions. In such cases remote work solutions might be the only alternative, but they may also be desirable in more general situations as they can be much more efficient in terms of cost and time, by avoiding transporting existing experts or fielding a co-located expert for each site [1]. Additionally, there was a surge of interest in the field due to the recent COVID-19 pandemic and the need for remote work that ensued [2]. For office work, traditional videoconference Computer Supported Cooperative Work (CSCW) solutions are often quite adequate, however, such solutions are not well suited for remote assistance for maintenance tasks, as spatial information is hard to convey through such mediums [3].

Augmented Reality (AR) / Virtual Reality (VR) technologies, however, are particularly well suited for conveying spatial information by using virtual objects that are referenced spatially, and are set to have a very significant impact on the field of CSCW [2], [4]. In recent years there has been a significant growth in interest for AR and VR technology driven by the proliferation of low cost Head-Mounted Displays (HMDs) and sensors that made these technologies affordable for the general public as well as cost-effective for a wider variety of applications, propelling a very expressive boom in research, commercial applications and general interest in the field [5], [6]. Now that the main engineering problems have been overcome [4] and the technology is more affordable, there is an opportunity for new applications of AR/VR technology to proliferate if properly designed to improve over the results for traditional solutions[1].

[1] https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off

Furthermore, remote assistance systems through AR/VR with support for asynchronous assistance, virtual replicas for communication or 3D model reconstruction of the local user's environment are not very common in research [6]–[8], which increases the relevance of works that make use of such approaches. Asynchronous collaboration has the potential to expand the time and cost savings of remote assistance even more, as it allows some level of parallel work and reuse of previous remote sessions. Specially if combined with independence of views between the local user and the remote user, which affords more autonomy between users, with 3D model reconstruction allowing a high level of independence of views with good quality. These characteristics synergize to use the remote expert's time more efficiently and there is currently no remote assistance system that combines all of them, which presents a good research opportunity to explore.

1.2 Objectives

The goal of this dissertation is to develop and evaluate a framework for remote assistance that makes use of VR and AR to connect a local user to a remote user with specialized knowledge to assist him in a maintenance task. The prototype should have the following characteristics:

- Allow the creation and sharing of a wide variety of annotations for non-verbal communication;
- Support asynchronous assistance through step-by-step instructions and storing them for later use;
- Provide an independent view of the local user's environment in VR to the remote user through 3D model reconstruction;
- Make use of virtual replicas to enhance communication of spatial information for assembly and object placement tasks.

1.3 Main Contributions

The following contributions were achieved:

- A remote assistance framework prototype combining all the aforementioned characteristics, filling an unexplored niche in current systems and prioritizing user autonomy;
- Evaluation of the prototype in a building maintenance setting, in comparison to traditional methods (pen and paper);
- A peer-reviewed publication resulting from the work of this dissertation which was published in the International Computer Graphics and Interaction conference [2].

Additionally, the work of this dissertation was presented at events such as *Students@DETI*, with a poster and demonstration videos of the prototype (see Appendix E - Students@DETI Content), and the *VIII Encontro Internacional da Casa das Ciências*[3] at which a live demonstration of the prototype was shown to secondary school teachers.

[2] http://www.icgi2023.ipt.pt/

[3] https://www.casadasciencias.org/8encontrointernacional/index.php

1.4 Document Structure

Apart from this introductory chapter, this dissertation consists of four chapters:

- **Chapter 2** introduces the problem and the technologies, followed by a review of the state of the art.
- **Chapter 3** describes the prototype and its development. It begins with the initial elicitation of requirements, followed by a description of the system workflow and implemented features. Lastly, it delves into the specifics of how these features were implemented, including the tools that were utilized.
- **Chapter 4** pertains to the evaluation of the prototype. It commences by defining the use case for the evaluation, followed by an explanation of the experimental setup, design and procedure. Afterwards, the obtained results are presented and discussed.
- **Chapter 5** is the final chapter, which will present the conclusions, along with a brief discussion of the main results and possible directions for future work.

CHAPTER 2

Remote Assistance Through Virtual and Augmented Reality

In this chapter, a brief introduction to the field of remote assistance, as well as virtual and augmented reality technologies will be presented, followed by the state of the art.

2.1 Remote Assistance

Remote assistance systems offer many advantages over in-person (co-located) assistance [9]. Co-location is not always viable or convenient for providing assistance or training for a task. An expert with specialized knowledge for accomplishing a given task might not be available, his knowledge might be required on a distant location, or even be required in too many locations in a short timeframe for him to realistically respond to [1]. In such cases, remote assistance systems might be the only option, but for other scenarios, they can also result in significant time and cost savings, which can make them a very attractive solution [1].

However, remote assistance can have its downsides, which need to be properly considered. Traditionally, remote assistance would be accomplished by text, audio or video calls, however, such methods are inefficient when it comes to maintenance tasks [3]. The local user has the benefit of a better view of the workplace which the remote user lacks, so the remote user might not be able to easily identify the problem and, additionally, his commands might not be well understood by the local user [10], [11]. Video and audio calls are inherently limited by their lack of depth-perception, communicating spatial information through speech is often challenging and confusing [12] while AR/VR is very well suited to convey spatial information since virtual objects are referenced in space. In addition, videoconference is not a good medium for non-verbal cues such as gestures (like pointing) and eye-contact, which is important to build trust and increase productivity.

AR/VR applications are a viable alternative to the traditional remote collaboration through videoconferencing and can overcome many of its limitations. In particular, the communication of gestures, gaze, annotations and user's emotions have already been demonstrated in previous works, which involve non-verbal cues that are often lost through videoconferencing [6], and can result in a more engaging experience, improving performance time and providing more intuitive and natural interaction [13].

Another important feature of remote assistance is that it can be used asynchronously, the information created for accomplishing a given task (gestures, annotations, etc.) may be stored and reused every time a user requires assistance or training, which results in further time and cost savings, which can be very significant [14]. And while video communication can also happen asynchronously (e.g. through instructional videos), an AR/VR application would entail all the aforementioned advantages over such traditional methods.

There is growing interest in remote collaboration through AR, as the number of published papers in the field has seen a very significant increase over the years, fuelled by the advancements in AR technology and development tools, as illustrated in Figure 2.1 [6].

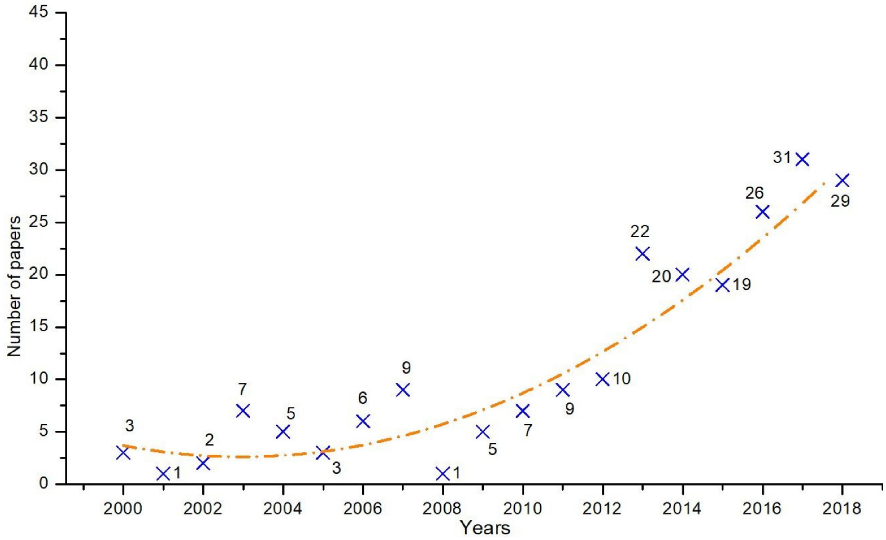


Figure 2.1: Number of research papers for AR/MR collaboration released over the years [6]. The X displays the number of released papers for that year, while the dash-dotted line shows the trend.

2.2 Virtual Reality

Virtual reality was defined by Schroeder in 1996 as "*A computer-generated display that allows or compels the user (or users) to have a sense of being present in an environment other than the one they are actually in and to interact with that environment*" [15].

Many other definitions exist in the literature and there is still not quite a standard definition [16] but the previous quote captures some of the most important characteristics of VR. A VR system should provide a high-degree of presence, interactivity and immersion.

Presence can be described as a psychological feeling of being in a distinct environment than the one the user is currently on, the Virtual Environment (VE) [17]. As for interactivity, it is a measure of the degree in which a user can influence the VE, how naturally and how fast.

Interactions in real-time and in a way that is natural to humans, such as moving a virtual object in 3D with a grabbing gesture and moving a hand, result in high levels of interactivity, which greatly influence presence. [18]

Note that even though commonly associated with HMDs, applications for other displays for computer-generated graphics can fit the aforementioned definition for VR if properly implemented, as long as they induce on the user a degree of being present on a virtual environment and have a high degree of interactivity, even if represented on a normal flat screen display. The definitions in the literature purposely try to be technology-agnostic, so as not to be limited to particular systems. However, not all applications that fit this criteria are good examples of VR technology, and this is where considering the level of immersion an application provides becomes an important distinction.

The more senses of the user are engaged in the application in a consistent manner, the more realistic the environment and the more the user is insulated from perceptions from outside the VE, the more immersive the experience [19]. It can be said that the applications that are usually most associated with VR and that have seen a major growth in recent years are in the high immersion category, the so called Immersive Virtual Reality (IVR), which typically makes use of HMDs with detailed environments and tracking of the movements of the body, usually through hand controllers and the headset.

Though VR systems have existed since the 1960s, they were bulky, heavy, low-resolution, and their cost was prohibitively expensive, which greatly limited its applications outside of research. However, between 2010 and 2016, the introduction to the market of consumer devices such as the Microsoft Kinect, Oculus Rift, Leap Motion Controller and the HTC Vive, marked the arrival of new generations of VR systems which are much more affordable and accessible [20]. As a result, the interest and research in the field has seen major growth, and the market for VR is expected to grow from US$16.67 billion in 2022 to US$227.34 billion in 2029, with applications in numerous industries such as entertainment, automotive, retail, gaming, healthcare, education, manufacturing, aerospace, defense[1].

2.3 Augmented Reality

A definition for AR that is often cited in the literature is as follows: "*An AR system supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world. (...) we define AR systems as having the following properties: combines real and virtual objects in a real environment, runs interactively, and in real time, and registers (that is, aligns) real and virtual objects with each other.*" [21]

So AR seeks to extend the real world with virtual objects, that are perceived to exist in the same physical space as other real objects, usually to add additional information about real world objects or represent information from digital systems in the real world. Similarly to VR this definition does not imply the use of specific displays like HMDs, a desktop computer flat-screen display or a smartphone screen are just as valid, and the same can be said for how

[1] *Virtual Reality Market Size, Share & Trends | Report, 2029*, `https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378` Accessed: 2023-04-04

the real world is presented to the user, which can make use of cameras or be observed through semi-transparent displays that superimpose the virtual objects.

VR and AR are very closely related technologies, both are used to represent and interact with computer-generated objects on a display, but while the former seeks to completely replace the user's reality with a VE, the latter attempts to only augment the real world with virtual objects. In fact, AR is often defined relatively to VR, and a complete spectrum of technologies can be defined between the two. Figure 2.2 contains one of the most cited representations of this spectrum of technologies, the so-called "virtuality continuum" [22]. On the left side we have environments composed solely by objects from the real world, which can be perceived directly or through a display, while on the opposite side we have environments completely formed by virtual objects. In-between these extremes, we have Mixed Reality (MR), which encompasses AR close to the left end of the continuum, and Augmented Virtuality (AV) on its symmetrical side of the continuum, which are distinguished by the balance of virtual vs real objects that form the environment.

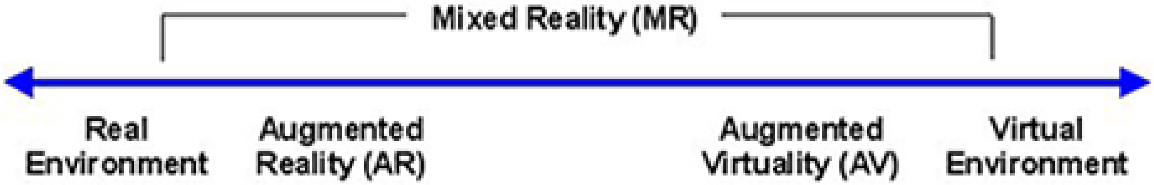


Figure 2.2: Simplified representation of a "virtuality continuum" [22].

Since 2016 there has been major growth in AR applications with the release of more affordable devices like the Meta 2, HoloLens, HoloLens 2 and Magic Leap One, which cost less than US$3500 and support Unity as an accessible SDK, as well as the increased awareness and interest from the public after the release of smartphone applications such as Pokémon Go [7]. Today, AR can be found in industries like gaming, manufacturing, media and entertainment, healthcare, retail, automotive, education and others, and the market is expected to grow from US$6.12 billion in 2021 to US$97.76 billion in 2028[2]

2.4 Previous Works in the Field

A few of the most relevant works in the field of remote assistance through AR/VR will now be presented.

In 2015, building on previous work that established the importance of the control of the viewpoint for both the local user and remote user [24], [23] Tait and Billinghurst were the first to create a system for remote assistance for an object placement task through AR that supported complete view independence through 3D reconstruction of the local user's environment, while also supporting communication through replicas of objects in the local user's environment.

[2] *Augmented Reality Market Size, Share & Trends | Report, 2029*, `https://www.fortunebusinessinsights.com/augmented-reality-ar-market-102553` Accessed: 2023-04-04

In their experimental design, the remote user was presented with a 3D reconstruction of the local user's environment created in real-time through a Microsoft Kinect and presented as fully textured models through a desktop interface, which he could pan and rotate with the mouse and communicate with the local user verbally and by placing replicas of real objects in the local user's environment. These replicas were presented to the local user through an AR HMD as transparent objects, while their physical object counterparts were shown to the remote user as transparent objects, along with a transparent representation of the view frustum of the local user. The views for both the local and remote user are presented in Figure 2.3 [23].

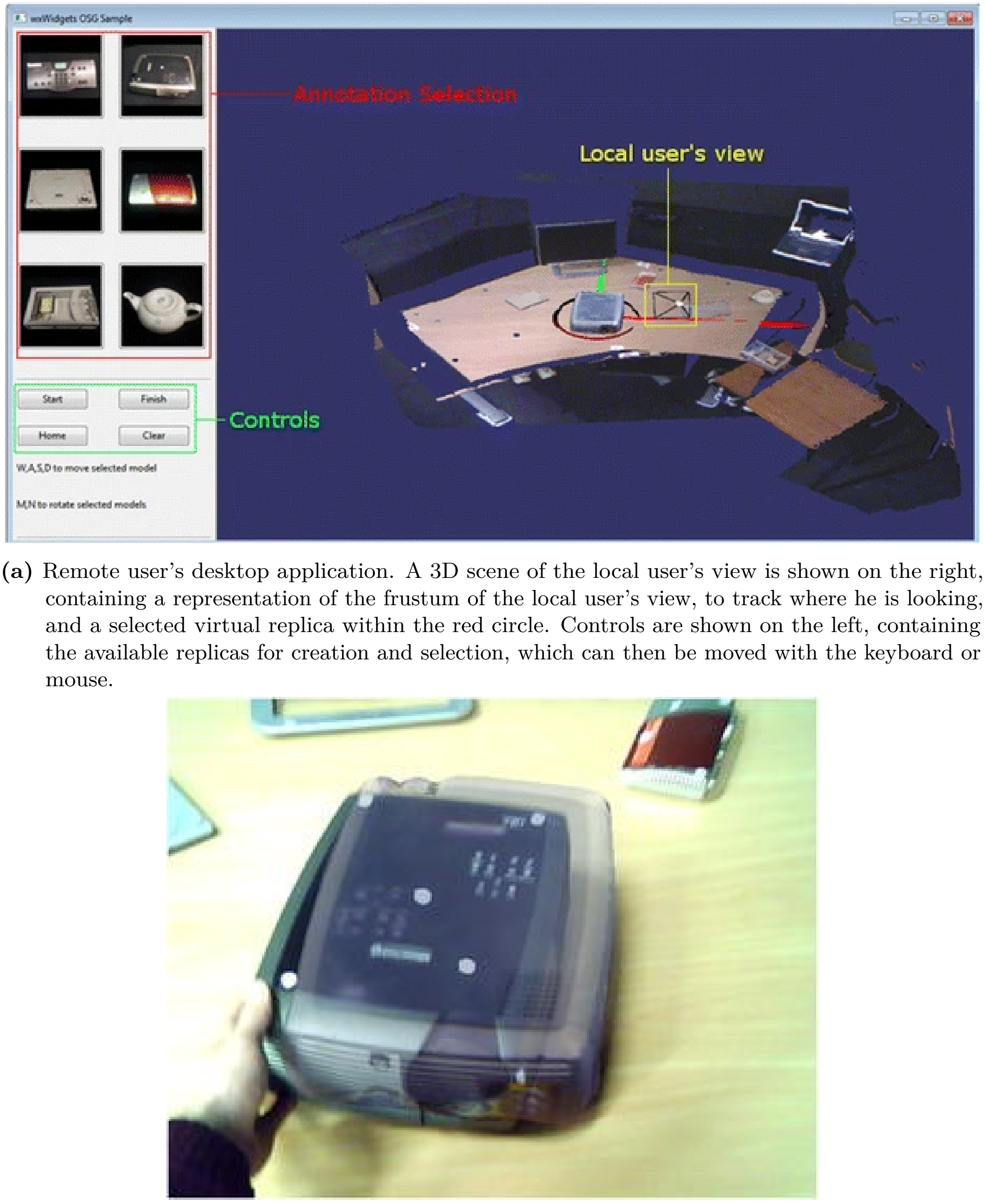
Their goal was to study how the degree of view independence influenced assistance, and by setting different levels of view independence they concluded through a user study that higher levels of view independence resulted in faster task completion and less time spent on verbal communication [23]. Their system was primarily designed with the intention of answering their research questions so the practicality of their system as an actual assistance tool is naturally limited, nevertheless, the main limitations will be pointed out for reference:

- Assistance was limited to object placement tasks only.
- The objects could be placed over a flat desk surface only - though the remote user's view was three-dimensional the object placement task itself was strictly two-dimensional.
- Assistance is strictly synchronous.
- Desktop interface for the remote user only - using an immersive VR interface for the remote user's view would result in more natural interactions and lesser cognitive load, allowing more focus on the task itself.

In 2015, Oda, Elvezio, Sukan, *et al.* created a system for remote training for assembly tasks that made use of virtual replicas to convey information on the task [25]. The relevancy of this work hinges on the fact that it was the first to demonstrate the merits of using replicas for remote assistance and there are still not many systems that follow this approach to this day. Furthermore, it remains one of the most cited examples of use of replicas for remote assistance in the literature to this day.

The local user wore a HMD for AR while the remote user used a HMD for either AR or VR and 2 tracked controllers, one for pointing and another for direct manipulations, allowing Six Degrees of Freedom (6DOF) manipulation and more natural interaction, while the relevant parts to assemble were also tracked in 6DOF and shown to the remote user as replicas with their equivalent pose in the physical environment [25].

The remote user had the option of switching between a VR view of the replicas which he could control freely and the AR view from the perspective of the local user, which therefore did not support view independence and was added for when it is necessary to view exactly what is happening on the local user's view, for example when the replicas do not correspond to the physical objects due to modification or damage. The replicas that represented the tracked physical objects were designated as virtual proxies, and the remote user could grab a virtual proxy, which allowed him to create a virtual replica of the proxy (without affecting



W,A,S,D to move selected model

M,N to rotate selected models

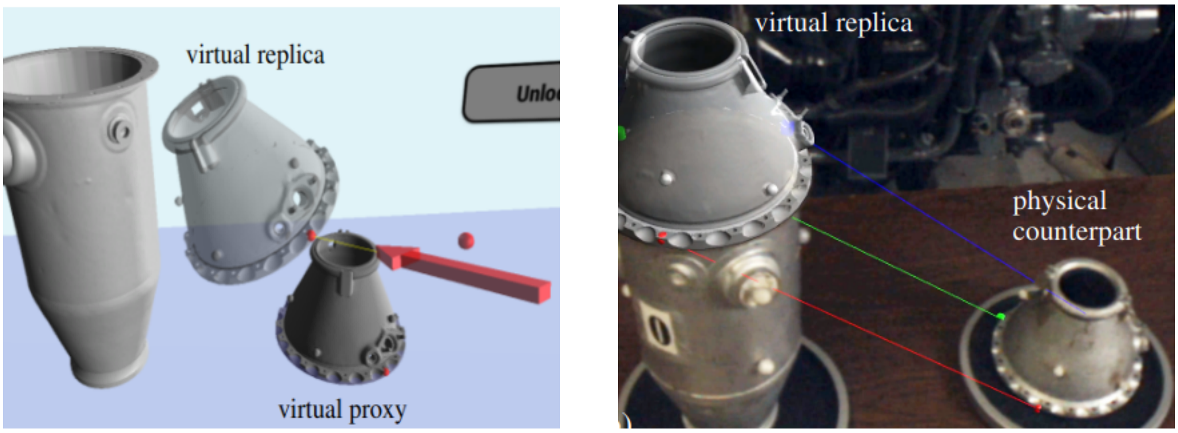
(a) Remote user's desktop application. A 3D scene of the local user's view is shown on the right, containing a representation of the frustum of the local user's view, to track where he is looking, and a selected virtual replica within the red circle. Controls are shown on the left, containing the available replicas for creation and selection, which can then be moved with the keyboard or mouse.

(b) View of the local user's HMD, while aligning a projector with the virtual replica positioned by the remote user.

Figure 2.3: User views for Tait and Billinghurst's system [23].

the proxy itself) that he could then place freely and would be shown in AR to the local user for communicating the assembly of the physical objects [25].

There were also 2 interaction modes for the remote user, which the authors designated as POINT3D and DEMO3D. For the POINT3D mode, the pointing device could be used to set 3 pairs of attachment points between any two replicas, represented by small solid cubes (for the local user) or spheres (for the remote user), to define their pose relative to each other, which would then have connecting lines to guide the assembly for the local user, and any replica that is pointed at would also have a 3D arrow pointing to it. As for DEMO3D mode, it was based solely on direct manipulations to set the correct pose for the replicas, and attachment points connected by lines were automatically placed connecting each virtual proxy and replica, to guide the local user. Figure 2.4 shows remote user views for both interaction modes [25].



(a) POINT3D interaction mode in VR mode. The remote user is using a pointing device to set attachment points between the virtual replica and virtual proxy (the small spheres), while the 3D arrow shows where he is currently pointing at and is also shown to the local user, to point to elements of the physical part.

(b) DEMO3D interaction mode in AR mode. The remote user sees the AR view as shown to the local user, showing the virtual replica in the final intended position and differently colored attachment points represented by small cubes, which connect 3 points of the virtual replica and the physical counterpart of the virtual proxy, to guide the placement of the physical part.

Figure 2.4: Remote user views for Oda, Elvezio, Sukan, *et al.*'s system [25].

An additional interaction mode was also designed and designated as SKETCH2D, in which the same AR and VR views would be shown to the remote user through a tablet instead of a HMD and interaction was based on screen touches. The remote user could pan and rotate the camera and sketch 3D lines that were drawn based on the point of intersection between the object in the touched pixel and the line that went from the center of projection. This setup was intended to be similar to other common implementations for remote assistance through AR at the time. Figure 2.5 shows a remote user view for the SKETCH2D interaction mode in VR [25].

The user study for the evaluation of the system showed that the direct manipulation of virtual replicas to convey the pose of objects was faster than direct annotations to complete an assembly task, showing the merits of this approach [25]. There are however some limitations for the implemented system as assistance tool, namely:

- Small degree of view independence - Since the AR view mode for the remote user was locked to the local user's perspective there was no view independence in this view mode, and while the remote user had some control over his view when in VR mode that view displayed only the replicas of the assembly parts with no information about the environment, reducing the benefits of view independence to improve the remote user's awareness.
- Mostly synchronous operation - This was acknowledged as a limitation during a pilot

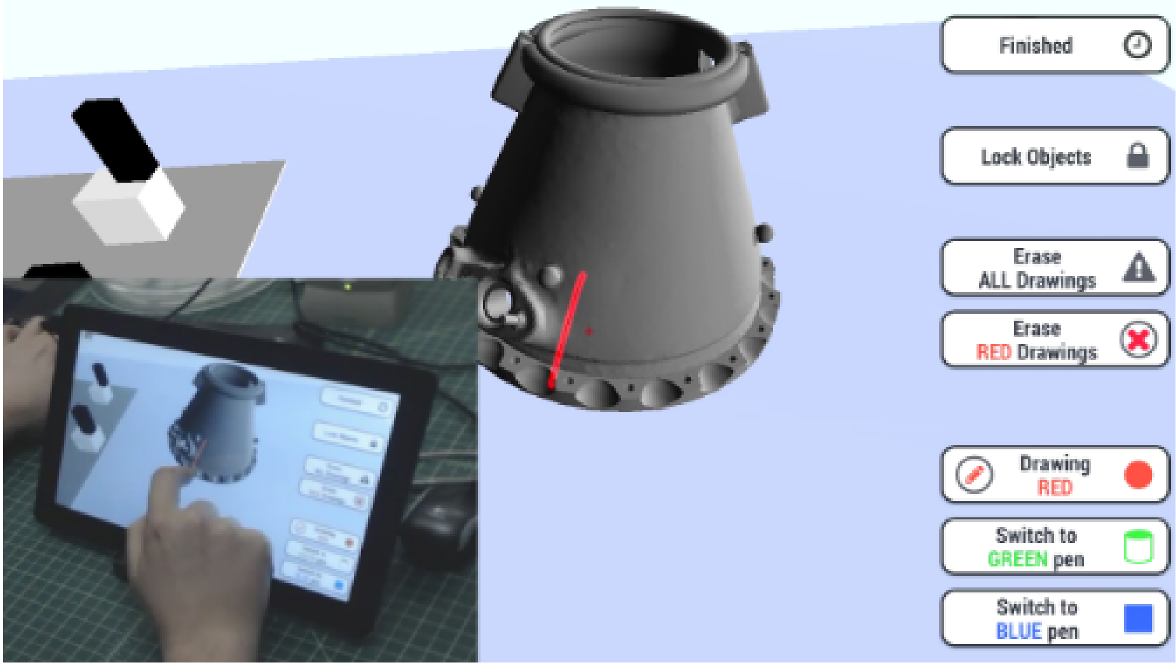


Figure 2.5: Remote user view in SKETCH2D interaction mode in VR for Oda, Elvezio, Sukan, *et al.*'s system [25]. The smaller picture shows the tablet being used by the remote user to sketch 3D lines, while the big picture shows a bigger view of the tablet's screen.

study which noted that on occasion the local user would start moving the physical part before the remote user finished his instructions, which interfered with completing the instruction. This led to partial support for asynchronous operation being implemented through a pedal that when pressed allowed the remote user to prevent tracking updates from the local user's physical objects from affecting his view [25].

- Limited annotation vocabulary - since the focus was on the use of virtual replicas, most of the non-verbal communication relied on them. There were other annotations available depending on the system configuration, but all were associated with the virtual proxy (and its physical part), such as showing an arrow pointing to a point on the part or sketching on the part, which reduces their flexibility.

In 2021, Calandra, Cannavò, and Lamberti set out to explore what they perceived as untapped potential in current research for remote assistance systems through AR. The significance of this work for the present dissertation is due to its focus on autonomy between the users in a remote assistance system, and demonstrating the advantages this autonomy can bring through the evaluation results of their prototype.

They recognized that the majority of systems require the remote expert to be available throughout the entire assistance session to accompany all the steps of the local technician, resulting in a one-to-one proportion in the use of the users' time. To make more efficient use of the remote expert's time, they studied the impact of a higher degree of autonomy between the users. To this purpose they created a system where users can open, close and restore old sessions as needed, maintaining AR content, which is organized in step-by-step instructions. In this way the remote expert can provide all instructions at the start of the session and leave the local technician to complete them, who can request further live assistance later if required. Additionally, for recurrent situations, the remote expert can re-use an existing session, and adapt the existing information as necessary, or the local technician can attempt to follow existing instructions before requesting real-time assistance.

The system makes use of a smartphone for the local technician, which through the 6DOF optical-inertial tracking of the ARCore library[3] displays virtual objects on planes set on the environment. The local technician sees the AR view of his workspace through the smartphone, while the remote expert receives a video stream of the local technician's smartphone view, on which he creates the virtual objects, while also communicating through an audio call. Many types of virtual objects are available, from hand-drawings and laser pointers, which are temporarily overlaid over the local technician's view, to arrows, spheres and instruction cards (containing text, images or animated GIFs) which are tracked in 6DOF. The virtual objects, organized into step-by-step instructions, and a video stream of each session are stored server-side, for later re-use. Images of some the available virtual objects can be seen in Figure 2.6.

Since their main goal was to prove the merits of an approach that emphasizes autonomy between users, the system was evaluated in comparison to a slightly adapted system that required the remote expert to be available throughout the entire session. These modes of operation were designated as Fully Assisted and Partially Assisted. The use case was for common operations with an industrial collaborative robot, the KUKA LBR iiwa 7[4], with the following tasks:

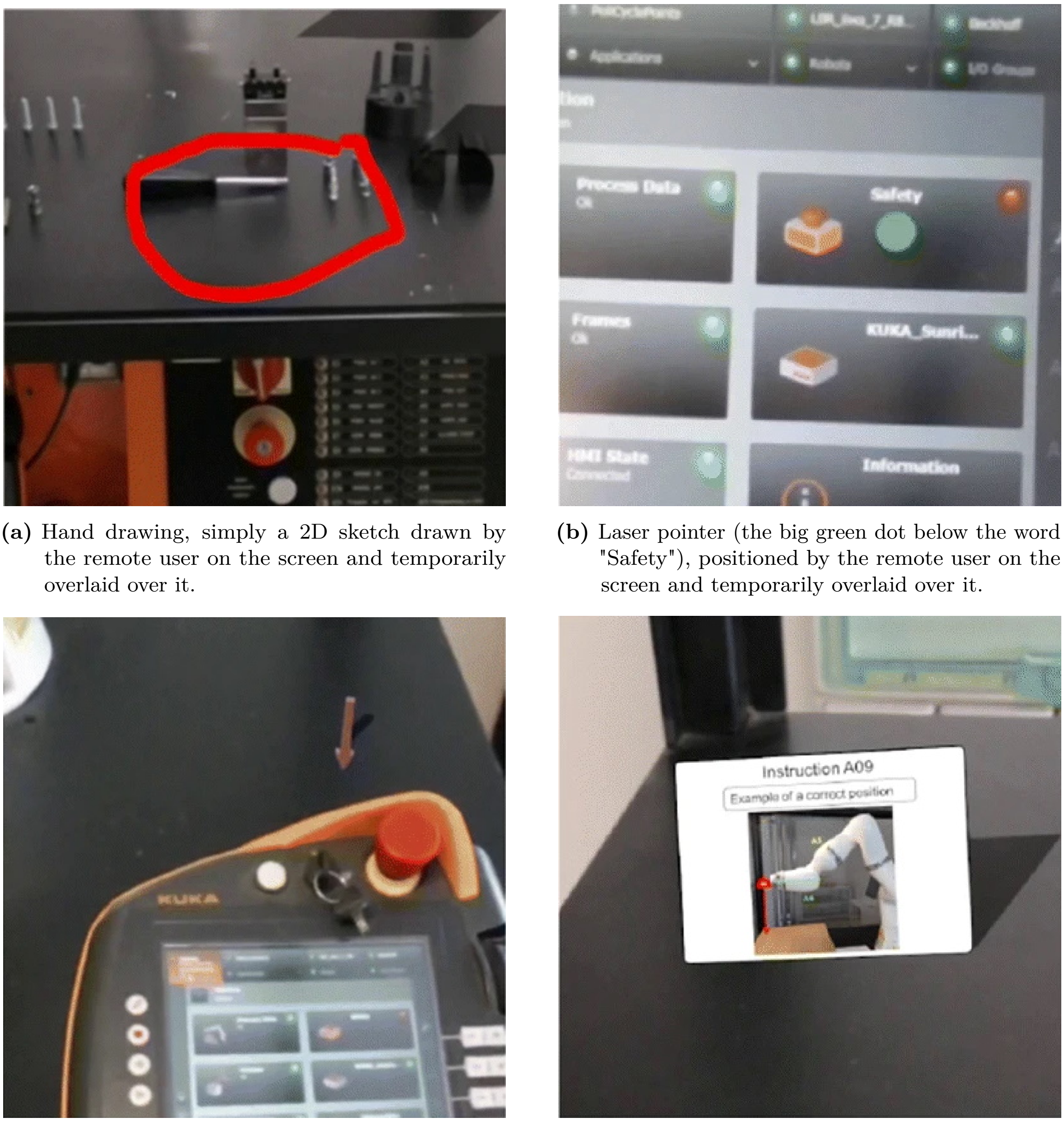
- Gripper Assembly (GA) - assembly of a robotic gripper, which typically takes considerable time due to the tightening of many screws.
- Load Data determination (LD) - a routine configuration task for calibrating the robot to handle a specific item, which requires many simple interactions with the robot's interface in the touchpad or jogging the robot axes.
- Emergency Recovery (ER) - recovery from an emergency stop, which in execution is similar to the previous task but requires some deeper understanding of the robot's operation, which are often unfamiliar to new users.

During the user study, the time that the remote expert was on call with the local technician was measured, as well as the time the technician took to complete the task. Additionally, some subjective measures were also assessed through a questionnaire.

The study's findings showed that the Partially Assisted approach significantly reduced the time required for expert intervention across all tasks, enabling operators to successfully complete procedures independently. These benefits were particularly pronounced in tasks involving numerous steps and substantial expert downtime, such as the Gripper Assembly task. As for the time taken by the local technicians to complete operations, it did not significantly differ between the two approaches, except for the Emergency Recovery task, which required an especially complex explanation, while the execution itself was relatively quick. This was

[3] https://developers.google.com/ar/

[4] `https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa`



(a) Hand drawing, simply a 2D sketch drawn by the remote user on the screen and temporarily overlaid over it.

(b) Laser pointer (the big green dot below the word "Safety"), positioned by the remote user on the screen and temporarily overlaid over it.

(c) 3D arrow, a 3D model positioned by the remote user with 6DOF tracking in the local user's environment.

(d) Instruction card containing a GIF, a 2D object positioned by the remote user with 6DOF tracking in the local user's environment.

Figure 2.6: Some of the virtual objects available in Calandra, Cannavò, and Lamberti's system [14], as seen in the AR view.

the only task where the Fully Assisted approach was preferred by the participants over the Partially Assisted approach, which was generally considered more useful and efficient, while better conveying the expert's knowledge without the pressure of keeping the expert waiting.

Overall, this study provided good evidence of the value of autonomy between users, so that the remote expert's time is used more efficiently without a significant penalty to the time the local user takes to complete the instructions. However, though the system was well suited to test their hypothesis, it had some limitations that could be improved upon:

- Lack of depth perception - only video-audio was shared with the remote expert. As for the local technician, he benefits from the AR annotations but they cannot be placed very precisely by the remote expert, as they are placed on a plane that depends on the position of the camera when they were created. This can reduce the benefits of spatially referenced annotations in AR significantly.
- Small degree of view independence between users - the remote expert cannot explore the local technician's view independently, as only video-audio is shared. Greater view independence could provide a very significant boost to the autonomy between users.
- Targeted devices emphasize accessibility over a richer experience - a smartphone is for the local technician and a web portal for the remote expert (which can run in any device with a browser). The use of HMDs would be more expensive but allow hands-free operation for the technician and richer interactions for both users.

Overcoming the first 2 limitations, in particular, could provide a much higher degree of autonomy between users, which could boost the benefits that were obtained over the real-time assistance approach, and allow real-time assistance to be completely optional.

At the end of 2021, Marques, Silva, Teixeira, *et al.* created a system for remote assistance for industry contexts, as part of an exploration into the methodology for development and evaluation of AR remote assistance tools [26]. It is of fundamental relevancy to the present dissertation as it intends to be a continuation of that research, and it is also a good example of a recent remote assistance tool that supports asynchronous assistance.

The system made use of AR through either hand-held devices or a HMDs for the local user and 2D views for the remote user which could be accessed and interacted through hand-held devices, computers or interactive projectors. The local user can take pictures of the regions of interest on their environment that they require assistance with, add 2D annotations to the picture with additional information such as the areas he requires assistance with, and then send it to the remote user, who can then add his own 2D annotations to the picture which is then sent back to the local user. Using the initial picture as a marker, the 2D annotations created by the remote user are then spatially registered on a plane on the local user's AR environment to be visualized through his hand-held device or HMD. There is voice communication along with a large lexicon of annotations, including text, sketching, arrows, notifications, sequential numbering for illustrating the order in which annotations should be interpreted, and the remote user can also incorporate 3D gestures through hand-recognition, suggest a change in region of interest or create step-by-step instructions using the created annotations which can also be stored for later use. The local user can also record and send video, which the remote user can then freeze and annotate at any frame, and the remote user also has the option of using footage captured during actual maintenance procedures to generate documentation. A representation of the system including screenshots of the user views and annotations is shown in Figure 2.7 [26].

Since it was developed through a participatory process involving industry experts, the prototype incorporated their insights, which resulted in a greater degree of refinement for actual use in the field compared to being strictly a research exercise. It is suited for both

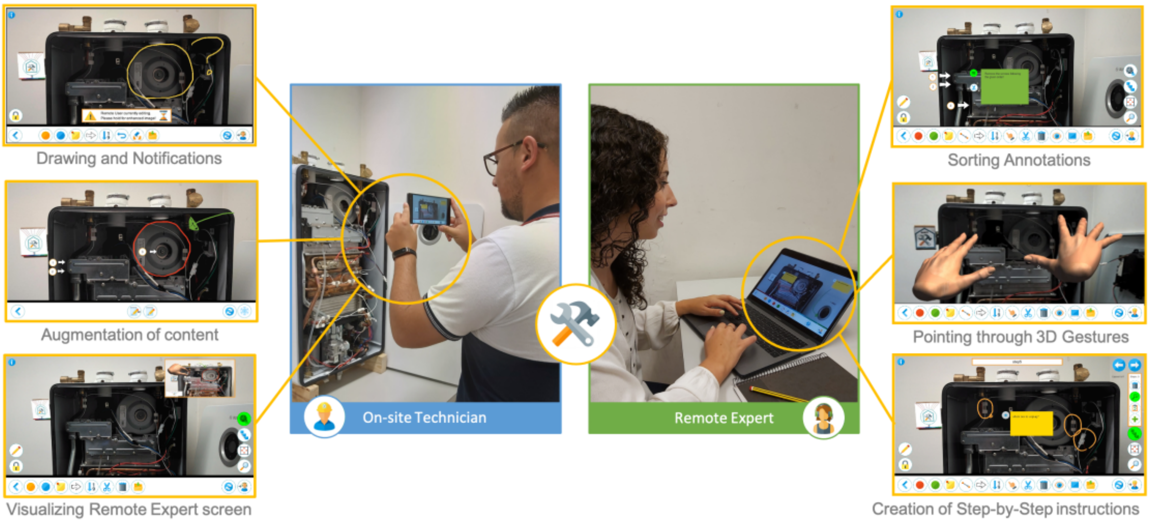

Drawing and Notifications
Augmentation of content
Visualizing Remote Expert screen
On-site Technician
Remote Expert
Sorting Annotations
Pointing through 3D Gestures
Creation of Step-by-Step instructions

Figure 2.7: Representation of Marques, Silva, Teixeira, *et al.*'s system [26]. The local user is shown on the left, using a hand-held device to see a AR view of his workspace, while the remote user is shown on the right, creating the AR content in the desktop application. Images and video can be shared between the users, which can then be annotated with 2D objects, which are shared and positioned in 3D in the AR view. 2D sketches, text boxes, arrows, and numbering (sorting) annotations are available, along with other features, such as notifying the local user when the expert is creating new content, gesturing by using hand-recognition, or organizing annotations into step-by-step instructions.

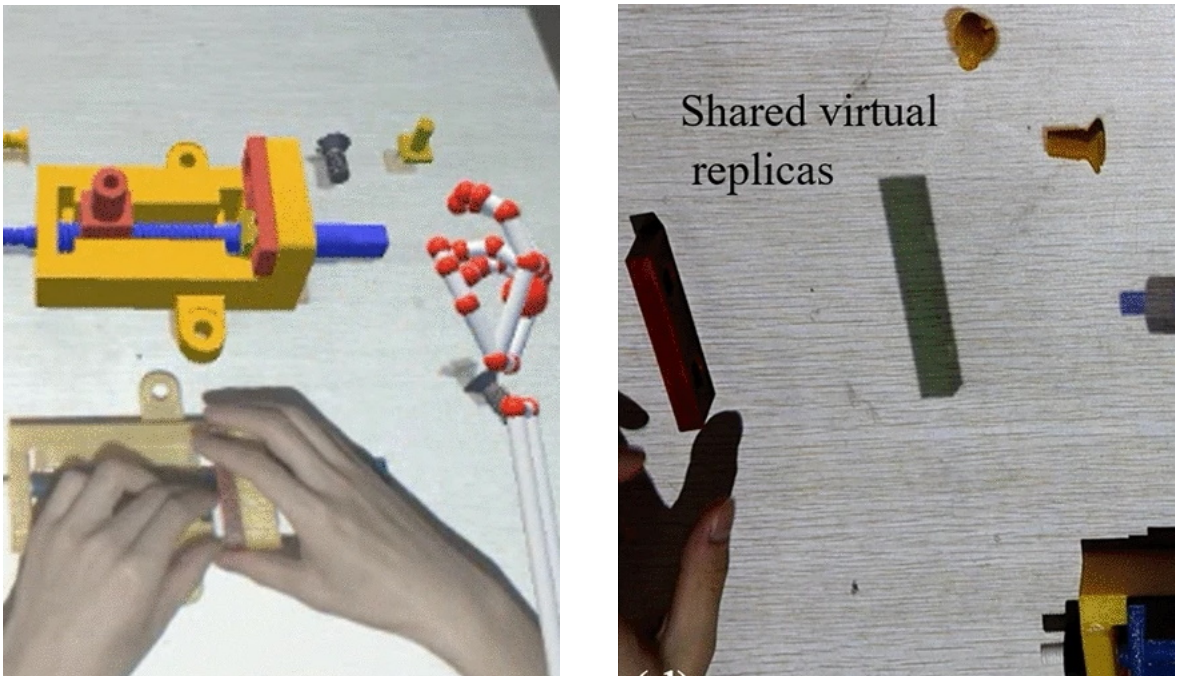
synchronous and asynchronous assistance, with features such as step-by-step instructions, documentation of previous maintenance procedures and text notes for asynchronous assistance, while notifications are specially useful for synchronous assistance. The main limitation of this prototype is that the remote user has no view independence and can only suggest changes in region of interest to the local user, for which the position may also be hard to convey. One of the changes proposed by the author for future work to address this limitation is the use of shared 3D models (such as the replicas in [25]) to convey the position of the new region of interest. Another approach would be to use a 3D reconstruction of the local user's environment for the remote user to explore in VR which would guarantee view independence and also allow conveying spatial information more efficiently through 3D annotations.

In early 2023, Wang, Wang, Billinghurst, *et al.* devised and evaluated a system using virtual replicas to assist on an assembly task, with a Spatial Augmented Reality (SAR) projector on the local user side and an immersive VR HMD for the remote user [27]. It constitutes a very recent example of a remote assistance system that shares some of the characteristics of the present dissertation's work, namely the use of virtual replica cues, AR for the local user, VR for the remote user and 3D scene reconstruction. In addition, it is the first remote assistance system to use gestures, replicas and avatars [27].

The users could communicate through audio and there were Reg Green Blue Depth (RGB-D) cameras on both sides: on the remote user's side for tracking his hands; and on the local user's side for reconstructing the environment and tracking his hands and body posture. For the remote user, replicas of all assembly task parts were presented, which could be interacted with through the recognition of a grab gesture to move them. As for the local user, the replicas were projected on the table where the assembly task is being performed. Three main configurations were tested: [27]

- The remote user sees the replicas of the parts and a 2D video of the local user's view presented on a vertical surface on his VR environment, while the local user sees the replicas projected on the table where he is completing the assembly task.
- The remote user sees the replicas of the parts and a 2D video of the local user's view presented on the same flat surface he is working on assembling the replicas of the parts on his VR environment, resulting in a similar view to what the local user sees with the SAR projection on the table he is working on, while the local user sees, in addition to the replicas, the tracked hands of the remote user for gesturing.
- The remote user sees a colored point-cloud of the local user's environment and the replicas of the parts, while the local user sees the same as in the previous setting.

All of these three configurations were also tested with and without a local user's avatar on the remote user's view, for a total of six configurations.

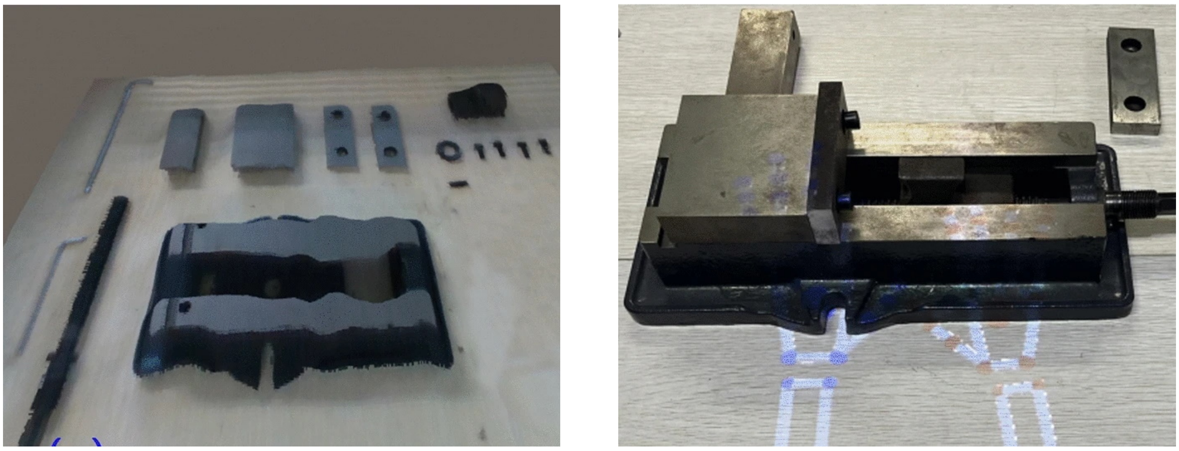


(a) Remote user view, with live-video of the local user shown on a horizontal surface (bottom), and the virtual replicas being assembled on the same surface (top-left) using gestures, with a representation of the remote user's hands (top-right).

(b) Local user view, with the SAR elements projected on the same table the user is working on the assembly task. Projected virtual replicas can be seen in the center and center-right, and their physical counter-parts on the center-left and top-center, respectively.

Figure 2.8: User views in Wang, Wang, Billinghurst, *et al.*'s system [27], for the configuration that shares the local user's view through a live-video shown on a horizontal surface.

The user study evaluated the system's usability, with good results, and the impact of using gestures and avatar cues, which had a positive impact on performance and communication, while social presence was improved for the remote user, and workload was not significantly



(a) Remote user view, showing the colored point-cloud of the table the local user is working on the assembly task and the respective parts.

(b) Local user view, with the SAR elements projected on the same table the user is working on the assembly task. The remote user's hands can be seen projected at the bottom, guiding through gesturing.

Figure 2.9: User views in Wang, Wang, Billinghurst, *et al.*'s system [27], for the configuration that shares the local user's view through a colored point-cloud.

impacted. The users also showed preference for the configuration of the system with the most non-verbal cues [27].

Some final remarks regarding the system as a remote assistance tool:

- Assistance is strictly synchronous - this impacts its usefulness as an assistance tool.
- Limited annotation vocabulary - this was acknowledged by the authors as a limitation and a sketch annotation feature was suggested for future work [27].
- No tracking of any physical assembly parts - unlike the implementation by Oda, Elvezio, Sukan, *et al.* [25], which results in a tradeoff between greatly simplifying the implementation and adding additional spatial cues for the remote user, this is, nonetheless, mitigated when displaying a 3D reconstruction of the local user's environment.
- No view independence, as the remote user either sees 2D video or a colored point-cloud that is based on the frame the local user is currently seeing, though the latter mitigates the lack of depth perception of a traditional videoconference approach.
- Fixed setup - the system is fixed for work on a table on the local user's side and thus not very flexible to be adapted for more assembly tasks. Adapting the used technologies for the local user's side could easily mitigate this, an AR HMD with depth sensors such as the HoloLens would allow the capture of the local user's view to accompany him, nonetheless, this seems to have been a deliberate decision on the part of the authors to make use of SAR as the authors cite literature that argues for its improved ergonomics, safety and competence when compared to HMDs [27].

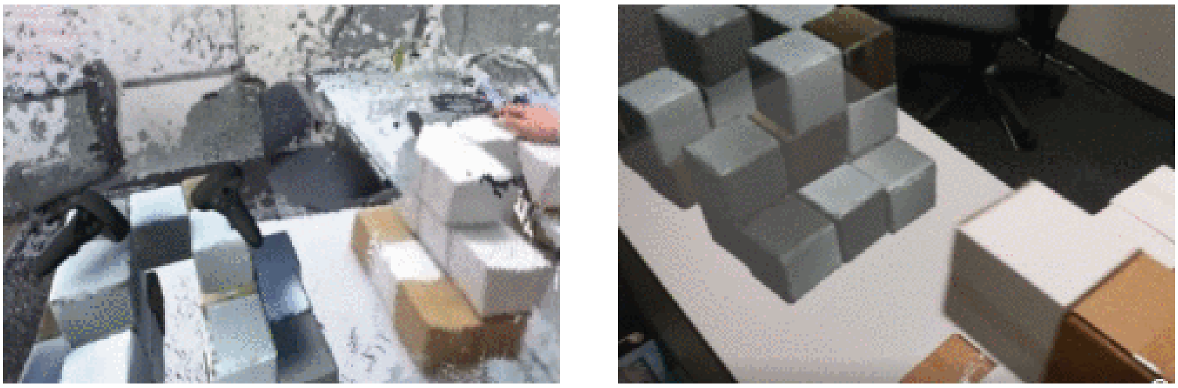
In February of 2023, Tian, Lee, Bai, *et al.* created a system notable for employing some innovative strategies to share the local user's environment with the remote user and also making use of virtual replicas for communication (among other cues). Similarly to the present dissertation, it makes use of 3D model reconstruction of the local user's environment, and is therefore relevant as one of the most recent examples of such an approach. Furthermore, it also introduces innovations that are not in the present dissertation but make it a good example of the state of the art of the methods for sharing the local user's environment.

It combines a static 3D model of the environment captured before the remote assistance session, and a live 3D point cloud for dynamic elements. A 3D model of the environment is more challenging to implement for real-time assistance, but provides better visual fidelity, so by capturing the model beforehand and using a 3D point cloud for dynamic elements, which is a faster method, the authors attempted to combine the benefits of both approaches.

In addition to making use of an HoloLens 2 for the local user and a Meta Quest for the remote user, the system employs a server to mediate all communication, three Azure Kinect Cameras to capture the point clouds and a Faro Focus Swift to create the static 3D model of the environment. Other than verbal communication, the system allows the users to see each other's body pose and hand motions, while the remote user can create 3D drawings and place virtual replicas for communication.

Evaluation of the system was performed for the assembly of a Soma puzzle cube (essentially a cube broken down into multiple parts that need to be carefully assembled in the correct order and orientation), made out of 7 pieces. The remote user had access to the solution and was tasked with guiding the local user using verbal communication and additional non-verbal cues, depending on the condition:

- Using 3D drawings;
- Using virtual replicas of the 7 pieces of the puzzle.



(a) Remote user's view, showing the local user's environment as a combination of its static 3D reconstruction and a colored point-cloud. The Soma cubes can be seen on the right, while their virtual replicas are shown on the left.

(b) Local user's view, showing the Soma cubes on the right, while their virtual replicas are shown on the left, to illustrate how they should be assembled.

Figure 2.10: User views for Tian, Lee, Bai, *et al.*'s system.

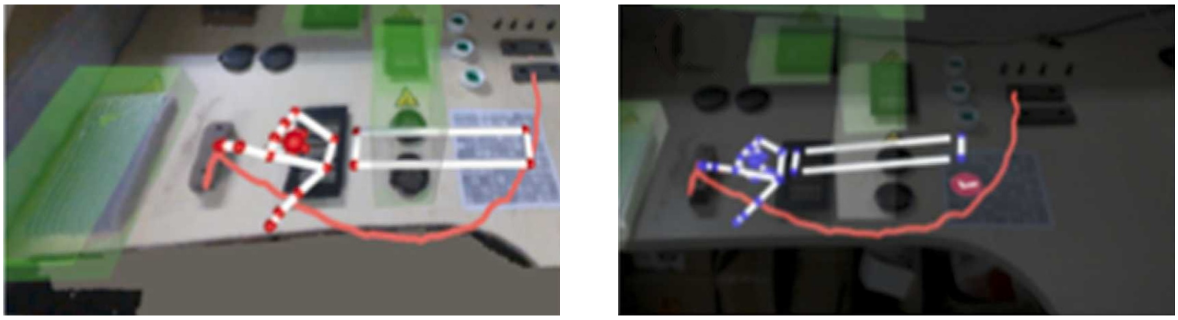
Results showed that virtual replicas improved task efficiency and reduced completion time and workload, compared to the 3D drawing condition, by improving understanding and reducing verbal communication time. Qualitatively, participants also favored the virtual replica condition, as it enhanced spatial and social presence, system usability, and reduced workload, reporting similar experiences in terms of social presence and task load for both local and remote users.

Overall the system provided a high quality shared view of the local user's workspace, with view independence between users, using a unique approach for sharing the local user's environment. This approach, however, requires significant setup, by setting up the multiple depth cameras and pre-scanning the environment model. In addition, it would be interesting to see further study for tasks that would benefit more from view independence, as the studied task did not require awareness of the workspace besides the local user's desk. Finally, the chosen conditions of the user study were chosen specifically to verify how virtual replicas would improve communication, and the results showed the merit of these communication cues.

In April of 2023, Zhang, Bai, Zhang, *et al.* created a system for real-time remote assistance and training, mainly intended for assembly tasks. The system makes use of immersive VR for the remote user side, using a HTC Vive Pro 2 HMD, to which a Leap Motion hand tracker is mounted to support interaction through gestures. As for the local user, he receives assistance through AR, through an HoloLens, also coupled with a Leap Motion, to support richer gesture interactions and recognize the position of the user's body. This work is a good example of the systems in the latest research in remote assistance through MR, using 3D scene reconstruction and an implementation that is based on very similar devices as the prototype of the current dissertation (with Microsoft HoloLens and HTC Vive HMDs).

The local user's environment is shared with the remote user as a colored point-cloud based on the frame the local user is currently seeing, so the remote expert cannot explore this view independently. The remote user guides the local user by sketching 3D lines, sharing his virtual hands and highlighting volumes of the environment. Sketching lines in 3D can be used for any purpose, but it was mostly used to highlight areas of interest and drawing arrows. Sharing the virtual hands enabled the remote user to communicate through gestures, while highlighting volumes of the environment was used to mark dangerous areas that the local user should avoid, with notifications for both users. Only the remote user interacts directly with virtual objects, by using a pointing gesture to sketch 3D lines, or a closed hand gesture to create planes to which he can add depth, thus allowing him to highlight a volume that marks a dangerous area. As for the local user, besides following visual cues, his interactions with virtual objects are limited to when he approaches a dangerous area, which changes in color and transparency the closer he is to it, to notify both users. Figure 2.11 and Figure 2.12 show both user views and an example of the notification system for dangerous areas, respectively.

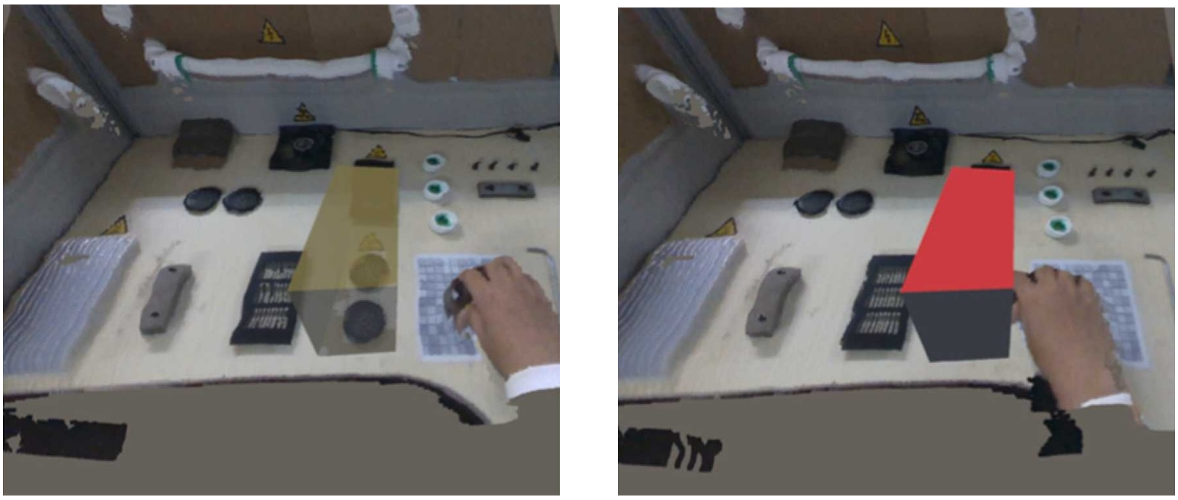
The system was evaluated for an assembly task containing risky areas that the local user needs to avoid while completing the task. The risky areas simulated common industrial hazards like radiators, cables, exhaust holes and mechanisms, which the remote user needed to highlight at the start of the assistance session, then proceed to guide the local user in the assembly while reminding him when he approached risky areas. In addition to the system's AR annotations, the users could communicate verbally outside the system, as they were seated physically close to each other, with a wall obstruction to block the view. Three conditions were considered:



(a) Remote user's view in VR, showing the local user's environment as a colored point-cloud. A representation of the remote user's tracked hand can be seen, which is using a pointing gesture to sketch 3D lines for an arrow. The transparent green areas are dangerous areas already highlighted by the remote user.

(b) Local user's view in AR, with the corresponding view for what is seen in (a). The local user sees all the same cues, including the remote user's hand representation, which the remote user can use not only for interacting with the application, but also for gesturing directly to the local user, such as by pointing to objects in the environment.

Figure 2.11: Remote and local user views for Zhang, Bai, Zhang, *et al.*'s system [29].



(a) Remote user's view, showing the local user working on the assembly task and an area that was highlighted by the remote user as dangerous (center). This area is initially green and transparent but has already shifted in color and transparency as the local user's hand (to its right) approached it.

(b) Remote user's view, showing the same highlighted area from (a) after the local user's hand has entered it. The area becomes completely solid and red to notify both users of the danger.

Figure 2.12: Zhang, Bai, Zhang, *et al.*'s system, showcasing the notification for dangerous areas [29].

1. 3DS - the baseline, making of the system's 3D sketching and sharing the remote user's virtual hands for gesturing;
2. 3DSA - which in addition to the previous conditions' communication cues, also allowed the remote user to highlight volumes to the local user;
3. 3DSAN - which in addition to the previous conditions' communication cues, also made use of notifications for when the local user approached or entered a risky area.

During the user study, objective measurements were taken for task completion time and risky operation errors (when a part of the local user's body goes inside a risky area), as well as subjective measurements for user experience (specially workspace awareness between users), preference and usability. In general, results improved the more visual communication were introduced between conditions, with 3DSAN having the best results, followed by 3DSA and then 3DS.

2.5 State of the Art Discussion

After a review of current remote assistance systems, a few of their key characteristics were identified that could present good opportunities for further research, which are still not very prevalent and have the potential to have a very significant impact on their usefulness:

- Support for asynchronous remote assistance;
- View independence between users;
- 3D reconstruction of the local user's environment;
- Support for replica annotations of physical parts for communication.

Each of these characteristics will now be defined in more detail, their importance will be discussed, followed by a brief overview of their prevalence in current research.

2.5.1 Asynchronous Assistance

Asynchronous assistance concerns the asynchronous creation and visualization of annotations, [7] either during a single remote assistance session or through the reuse of instructions from previous sessions. It is important to reduce the time spent by the remote user on assistance sessions, either during sessions, as the local user can focus on a task already sent by the remote user while the remote user creates the next set of instructions, or by reusing the instructions of previous sessions [14].

A survey by Sereno, Wang, Besançon, *et al.* has verified that remote asynchronous collaboration through AR is not very common in research [7]. Asynchronous operation is important as it allows the local and remote user to work in parallel and re-use existing instructions, and again the present dissertation falls into this category that is still not very explored in research, which further enhances the relevancy of this work.

One additional insight from Sereno, Wang, Besançon, *et al.*'s review is that 69% of published papers between 1996 and 2019 for remote collaborative work in AR use asymmetrical technologies for the remote and local user when their roles are also asymmetrical and vice versa, and the most common choice of asymmetric technologies is AR for the local user and VR or a desktop screen for the remote user [7]. In the context of remote assistance, the collaboration has asymmetric roles, as the local user has physical access to the workspace and tools, while the remote user has the knowledge required to complete the task. Hence, the design decision of using AR for the local user and VR for the remote user for this dissertation is in line with current research trends.

2.5.2 View Independence

View independence is an important topic for remote assistance through AR/VR. In 2013, Lanir, Stone, Cohen, *et al.* were one of the first to demonstrate the importance of each user controlling the point of view, citing previous research and its limitations.

Previously, some systems presented static views to the remote user, which are not very flexible as they are not always able to get a full view of the workspace, which might contain multiple planes and small items to focus on, and in some cases the location of the workspace itself can change. Others made use of dynamic views through multiple static cameras, head-mounted cameras for the local user or moving the camera through robotic arms which are controlled by the remote user or automatically pointed to the local user's hands.

Other approaches include showing the local user's view to the remote user through 360º cameras or a 3D representation of the environment. Using 360º cameras usually results in a view that supports rotations but no translations, while 3D representations provide some depth perception and possibly a higher degree of view independence, possibly allowing the remote user to explore the environment independently.

Another consideration is that view independence is specially relevant when combined with some degree of asynchronous assistance for creating and sharing annotations. It allows the local and remote user to work in parallel on different regions of interest on their views for a single session, or use the local user's environment information that was stored from a previous session to create new annotations for later use. Therefore, view independence can allow some level of parallel work and greatly increase the autonomy between the users. Additionally, greater autonomy between the users can result in more efficient use of a remote expert's time, as shown in [14].

Wang, Bai, Billinghurst, *et al.* conducted a review on publications for remote collaboration using AR/MR for the period of 2000-2018, considering 211 papers. One of the trends identified by the author is that the most used method for presenting information about the local user's environment to the remote user is still video/audio, representing more than three-quarters of current research, followed by 3D scene reconstruction environments and then 360º cameras. Therefore, the use of 3D scene reconstruction is still far from prevalent in research, which enhances the significance of the present dissertation.

2.5.3 3D Reconstruction

There are multiple approaches in current research to create a 3D representation of the local user's environment. Given the prevalence of asynchronous assistance in research, the most common approach for creating a 3D scene representation is using point-clouds, as seen in [27] and [29]. Actual 3D reconstruction to create models/meshes of the environment using the point-cloud is relatively uncommon. After a review of the literature it seems that even among the subset of prototypes identified in [6] as using 3D scene representations, the ones that create 3D models of the environment are in the minority, with [28] being one of the more recent examples. This is due to 3D model reconstruction being more computationally expensive, which makes it more challenging to use in real-time assistance [8], often resulting in slower update times. However, 3D reconstruction can provide the best quality view [8], and possibly richer interactions with the environment in AR/VR, as collisions with the model surfaces can be easily computed.

Given the focus of this work on asynchronous assistance, the real-time constraints are lifted, which presents a good opportunity to make use of 3D reconstruction as slower algorithms can be used. Given the advantages of 3D reconstruction and the fact that it is a subset of a subset of current research, making use of this approach increases the relevancy of the current work.

2.5.4 Replica Annotations

Replica annotations are a type of non-verbal cue in AR/VR remote assistance systems that uses virtual representations of objects that exist in a physical environment (usually the local user's environment). They can represent tools, or objects that the user needs to move, such as equipment parts.

There are various applications of replicas in the literature for remote assistance through AR/VR, and they are an effective way to communicate for assembly and object placement tasks, they do, however, have the downside of requiring a pre-stored database of models to be used, reducing its flexibility [14], [23], [25], [27]. In some systems, the position and orientation of the replica follows the real object's position, similarly to a digital twin, though tracking can be challenging and have problems when the physical object is occluded [25].

In Wang, Bai, Billinghurst, *et al.*'s aforementioned review on publications for remote collaboration using AR/MR, the types of non-verbal cues used for the implementation of each work were categorized by the authors. Their findings show that virtual replicas are not a common cue-type in research, and the proportion of papers that did make use of it has remained relatively stable over the years since 2003. Therefore, there are still a lot of opportunities to explore this particular area of remote collaboration through AR/MR, which increases the present dissertation's relevancy.

CHAPTER 3

Prototype Development

In this chapter, the prototype and its implementation will be described, and its design decisions discussed.

3.1 Application Scenario - Remote Assistance for Maintenance

Having a well-defined application scenario is essential to guide application design through each stage of development. Therefore, to keep development focused, the chosen use case for the system prototype was maintenance for an office building, though it might have potential to be viable as a remote assistance tool for other scenarios. There are many systems in an office building that require maintenance, from the electrical system, plumbing, Heating, Ventilation and Air Conditioning (HVAC), etc. All of these require regular maintenance actions to keep working in good condition and avoid breakdowns, and for any building of significant size the number of tasks can quickly become overwhelming and hard to keep track off. In such cases, adequate management of the maintenance process becomes essential to get good results and keep costs down, which are usually benchmarked through Key Performance Indicatorss (KPIs).

In maintenance management, there exist many taxonomies to categorize maintenance types and the terminology keeps evolving, but maintenance actions usually fall into one of these types [30], [31]:

- Preventive maintenance;
- Reactive maintenance.

Preventive maintenance comprises all maintenance actions with the goal of avoiding equipment breakdowns before they actually occur. These actions are usually planned beforehand or based on pre-defined criteria and include tasks such as routine inspections, lubrication and exchanging expendable parts (such as filters). Though often overlooked in some (usually smaller) organizations, a focus on preventive maintenance is pivotal to reduce cost and equipment downtime, and a long life expectancy of the equipment.

Reactive maintenance, on the other hand, are maintenance actions performed after a full, or partial, equipment breakdown to restore it back to full working condition. It is the simplest maintenance strategy that requires the least planning and, though occasional breakdowns are unavoidable, a heavy focus on these maintenance actions has the aforementioned disadvantages compared to preventive maintenance.

For deciding the type of maintenance actions for the application's use case, it was taken into account that corrective maintenance actions present some challenges to their testing on a real-life scenario when compared to preventive maintenance actions. Simulating corrective maintenance actions would mean equipment of some complexity would have to be available, in addition they often require expert knowledge of the equipment being serviced and how its subsystems interact so that the problem can be promptly diagnosed and fixed. Therefore, to simplify the application's testing procedure, an emphasis was placed on preventive maintenance actions.

The simplest maintenance actions in preventive maintenance are usually inspections, specially if relating to simple equipment or verifying if a building's section is following fire safety and evacuation regulations, or other simple tasks. In maintenance management and most Computerized Maintenance Management Systems (CMMSs), these are usually expressed as Work Orders (WOs), containing maintenance checklists, where each element is to be assigned an Okay/Not-Okay (OK/NOK) tag, possibly also registering a value from an equipment (such as reading oil pressure on a dial) or adding some further comment about the equipment's state.

Thus, the application's use case will relate to an inspection round of a small section of a building, defined as a checklist or series of instructions. The resulting application can then be used as a remote assistance tool for an expert to provide one-time instructions for an inexperienced technician that is unfamiliar with a particular inspection round, or as a training tool for local technicians to re-use the instructions until they become familiar with the task.

3.2 Elicitation of Requirements

In this section, the initial requirements for the system will be described, which are provided as a reference, as they were later exceeded in some regards during development. It is generally established among both researchers and industry professionals that a good definition of requirements is essential to the success of a software project [32], [33]. For this project's prototype, the chosen requirements elicitation technique was brainstorming sessions with elements of the research group in which this dissertation work is inserted (VARLab[1]), making use of their expertise in the field of AR/VR and past experience with previous projects.

3.2.1 Personas

As an alternative to a simple list of requirements, personas are an alternative method of expressing user needs and goals, initially defined by Cooper in 1999 [34]. A persona is a

[1] `https://sites.google.com/view/varlab/home`

fictitious character that represents an archetype of the users of an application, encapsulating a group of users that share the same main characteristics, goals and needs [34]. In line with Human-Centered Design (HCD) practices, they provide many benefits and are invaluable to ensure that the focus remains with the future users of the application for guiding design decisions and challenging the initial assumptions of the developers [35].

The following personas were defined:

Name: Zachary
Age: 22
Occupation: Junior Maintenance Technician

As an inexperienced maintenance technician in a new position, I want to receive assistance even when an expert is not available on-site, so that I can be more productive by overcoming problems that pop-up during the course of my duties and keep learning about the relevant activities of my job position.

Name: Christopher
Age: 45
Occupation: Expert Maintenance Technician

As an expert in the maintenance tasks of my company, I want to be able to easily provide guidance to new recruits even when I am not on-site, so that I can use my time more efficiently and be available in the facilities where I am most needed.

3.2.2 Requirements

Functional Requirements

The following functional requirements were defined:

- The system should be composed by 2 parts: an AR side for the local user to visualize instructions as virtual objects and a VR side for the remote user to author the instructions, on a different location with no direct access to the local user's environment.
- The instructions should be composed of 3D referenced virtual objects (annotations) organized as step-by-step instructions to be visualized sequentially.
- The following types of annotations should be available:
  - Areas of interest, represented as circle highlights;
  - Arrows;
  - Comments;
  - Replicas.

- On the VR side, a 3D reconstruction of the local user's environment should be presented. This was suggested to be implemented either through an existing point-cloud of a specific location or by acquiring a new point-cloud through a 3D mapping device, such as the Leica BLK[2]. This mapping of the local user's environment is considered part of the initial configuration of the system for it to be used in that space, and not part of the system operation itself.
- Instructions authored in the VR side should be made available later for visualization on the AR side by sharing a file, which can be re-used as many times as necessary. Therefore, the process starts with the VR side, ends in the AR side, and is asynchronous, so the local and remote user do not need to use their respective applications simultaneously.
- The annotations created on the VR side should be referenced on the AR side in the real-world equivalent of their position in the VE. This was suggested to be implemented by referencing the virtual objects in the AR side through a physical Quick Response (QR) code, which would have a virtual counter-part on the VR side that would be positioned on the correct position by the remote user in a configuration phase, to align the physical and virtual environments.
- Target either the HTC Vive or Meta Quest 2 platform for the VR side.
- Target the Microsoft HoloLens 2 platform for the AR side.

Non-Functional Requirements

Though not explicitly defined in the brainstorming sessions, the following non-functional requirements emerged from the particularities of AR/VR technologies and the chosen platforms:

- **Performance** - both the Microsoft HoloLens 2 and Meta Quest are mobile devices with limited processing power. Therefore, some care should be taken to ensure that the rendering of the application is not too demanding for these platforms and a consistent framerate and low latency is maintained throughout its use, as these would result in a poor User Experience (UX) and are some of the main causes of cybersickness.
- **Interoperability** - the system comprises two distinct applications on different platforms, one for the AR side and another for the VR side, each with their own Software Development Kit (SDK), so ensuring they operate effectively between them is essential.
- **Portability** - initially, it was suggested that development of the VR application could start on the HTC Vive, and on a later phase of development the Meta Quest 2 could then be targeted. Some degree of portability is therefore desirable, not only in case other devices are targetted, but also because it could allow some code to be reused between the VR and AR applications, expediting development. Moreover, it has historically been a problem with VR development that SDKs for older devices are no longer supported on modern platforms, so reducing the dependency on specific devices can make the system more future-proof.

[2] https://leica-geosystems.com/products/laser-scanners/scanners/blk360

3.3 Prototype Description

Having defined the initial requirements for the system, the actual resulting implementation will now be described. As illustrated in Figure 3.1 the system comprises two distinct applications on distinct platforms. The local user uses an AR headset (Microsoft HoloLens 2), while the remote user uses a VR headset (HTC Vive). The VE where the remote user is immersed is a 3D model reconstruction of the local user's environment, generated from the depth sensor data of the HoloLens device.

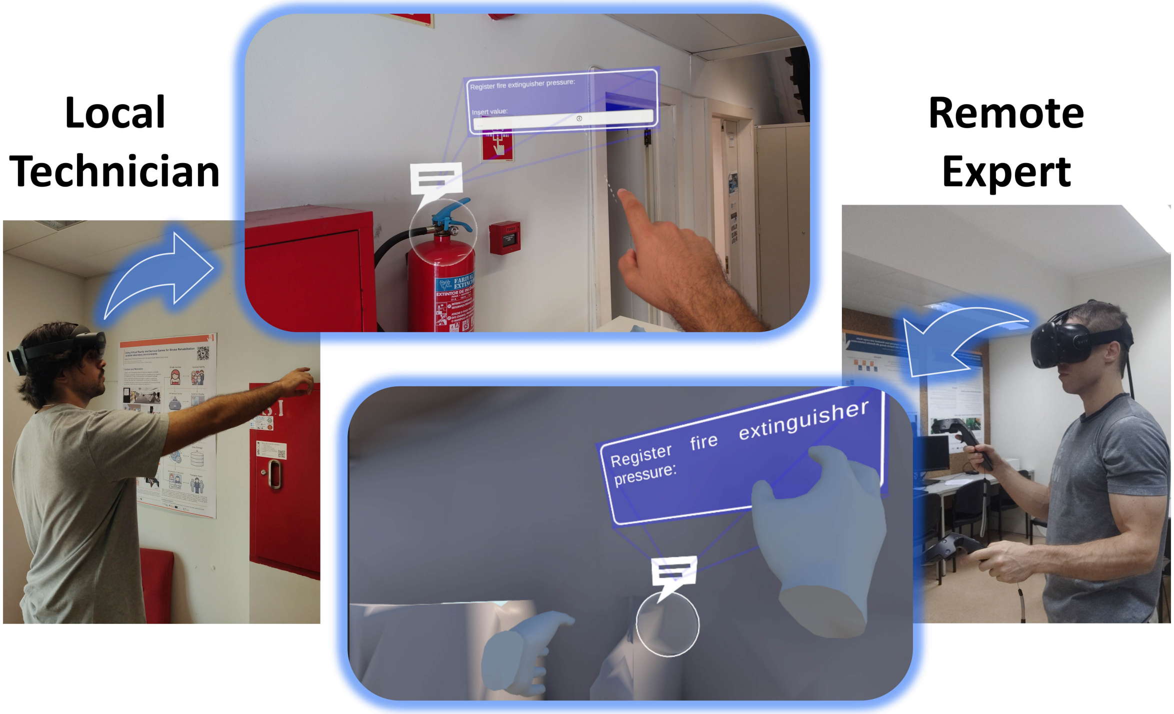

Local Technician
Register fire extinguisher pressure:
Insert value:
Remote Expert
Register fire extinguisher pressure:

Figure 3.1: The AR (left) and VR (right) applications that comprise the system, used by the local technician on a Microsoft HoloLens 2 and the remote expert on a HTC Vive, respectively.

In this section, the system's features and its operation from the user's point of view will be described, starting with the general workflow of the system across both applications, the functionality that is common between them, and then addressing the AR and VR sides separately, in more detail.

3.3.1 System Workflow

A full remote assistance cycle involves multiple steps across both applications, performed by different users on different locations. As illustrated in Figure 3.2, this involves 3 stages, starting with a local technician in AR to scan the environment, a remote expert in VR to author the instructions and finally a local technician again to see and execute the instructions. A demonstration video of the full cycle is available in YouTube[3].

[3] https://www.youtube.com/playlist?list=PLpEPS1zh4byfnBfbsgQwNtfK4EKOWuknY

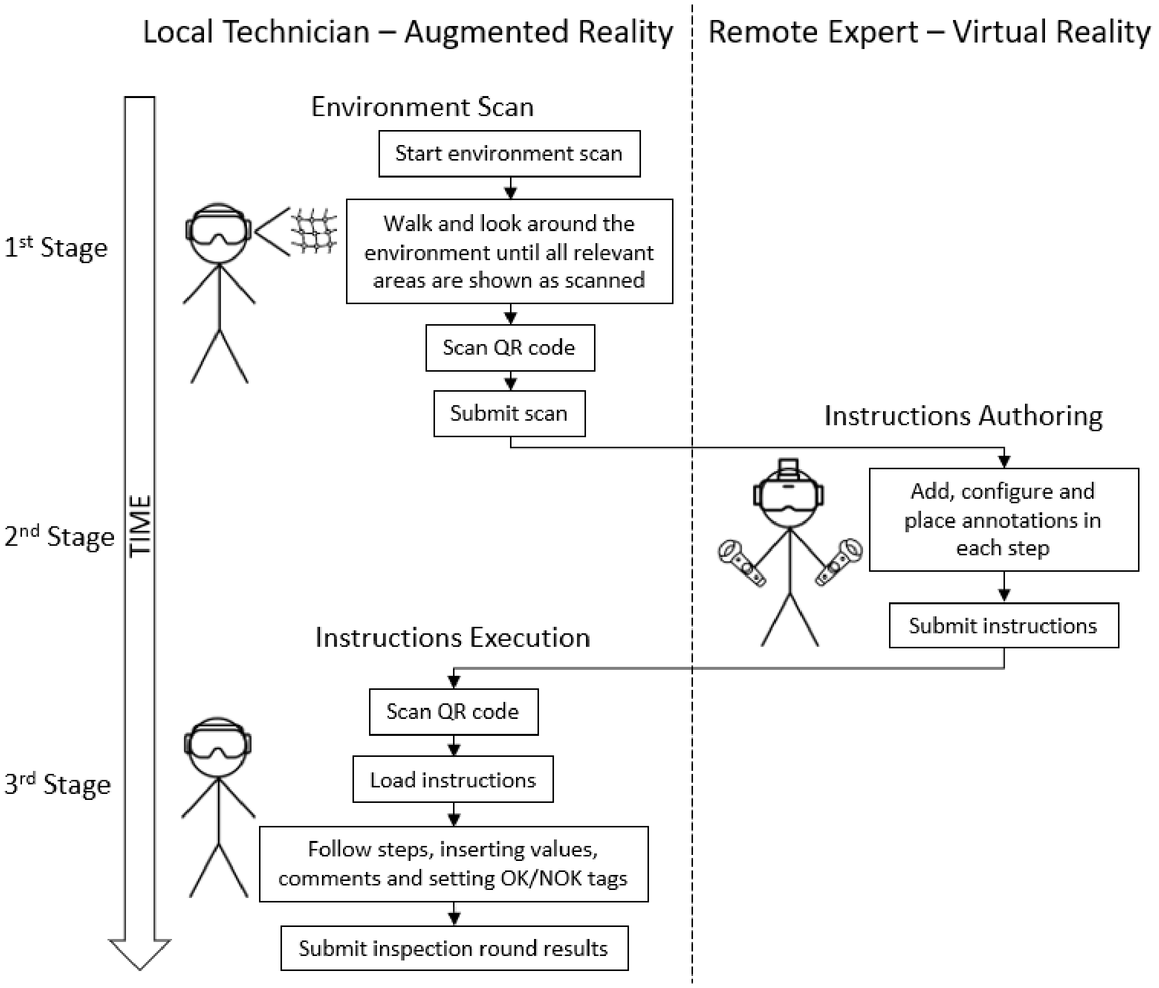

Local Technician – Augmented Reality
Remote Expert – Virtual Reality
TIME
1st Stage
2nd Stage
3rd Stage
Environment Scan
Start environment scan
Walk and look around the environment until all relevant areas are shown as scanned
Scan QR code
Submit scan
Instructions Authoring
Add, configure and place annotations in each step
Submit instructions
Instructions Execution
Scan QR code
Load instructions
Follow steps, inserting values, comments and setting OK/NOK tags
Submit inspection round results

Figure 3.2: Diagram of the system workflow, involving a local technician and a remote expert on different sites, using distinct devices, operated sequentially at different times.

A QR code is used to reference the virtual objects in AR, so to ensure consistency, in the 3rd stage it needs to be located in the same position and orientation as it was during the 1st stage. More detail about the referencing method will be provided later.

At the end of each stage a file is produced, to be used later for the next one or to store results, so all of these steps happen asynchronously. That is, at no point is it required that both users be using their respective applications simultaneously, and each subsequent stage can happen at any time after the previous one. The only requirements are that the QR code has a consistent location and orientation, as already stated, and that the initially scanned environment has not changed in a way that could render the instructions invalid.

In addition, after the 1st stage, subsequent stages can be repeated as many times as necessary using the produced file. So the process can branch into multiple sets of instructions, for the remote expert to react to new situations in known environments, or the same instructions can be used multiple times, for recurrent situations, periodic maintenance or training scenarios. For periodic maintenance, instructions could even be created for each maintenance plan and the results (such as readings and completion status of each task) be automatically sent to a

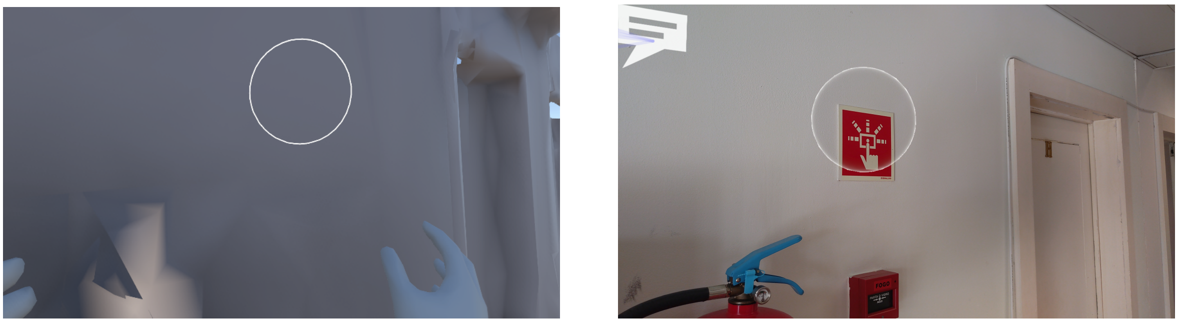
CMMS instead of inserted manually.

3.3.2 Annotations

Annotations are the virtual objects used to convey textual and non-verbal cues between the remote expert and the local technician, which are grouped into step-by-step instructions. They are created in the VR side by the remote expert, and visualized on the AR side by the local technician, so can appear in both applications. They have, however, slight differences for each application that were necessitated by the distinct characteristics of the different devices and tools that were used to produce each application, and slightly different functionality where appropriate. The two applications have different rendering pipelines and interaction toolkits, so each annotation type had to be configured independently for each, as for their differences in functionality they vary by annotation type and will be examined in the following sections.

Areas of Interest

These annotations are used to highlight specific areas of interest. They are defined by the outline of a sphere, to avoid occluding the environment and other annotations. This approach was favoured relative to using 2D circles, as was initially proposed, so that the annotations have the same representation from all angles and can be used to consistently highlight a certain volume in the environment. Figure 3.3 shows a area of interest annotation in both applications. On the VR side the user can move and resize these annotations, while on the AR side they can only be visualized.



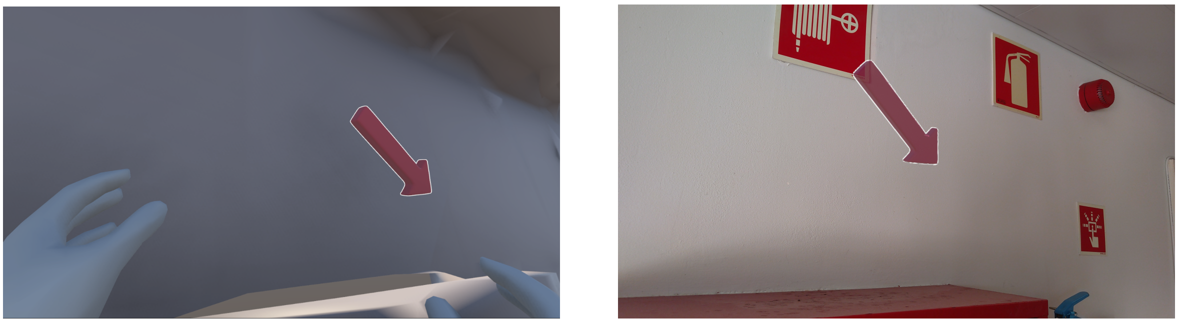
(a) VR side.

(b) AR side.

Figure 3.3: Area of interest annotations as rendered by both applications.

Arrows

Arrow annotations point out specific objects or define directions. Figure 3.4 shows an arrow annotation in both applications. They can be moved, reoriented and resized on the VR side, while on the AR side they can only be visualized.



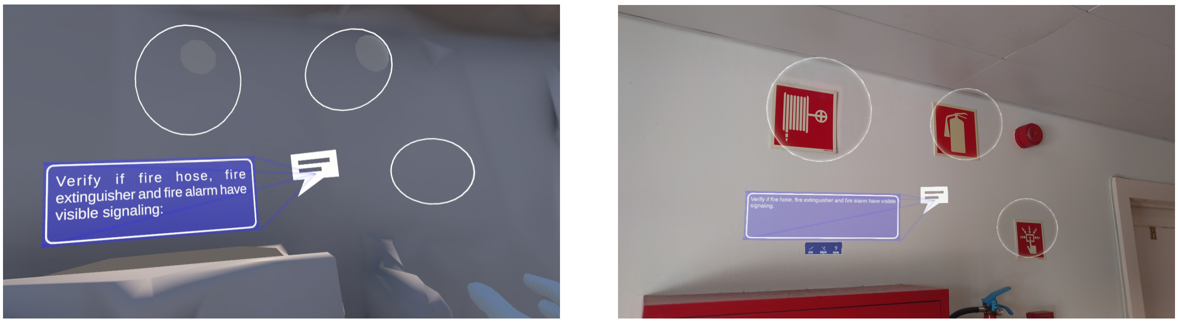
(a) VR side.

(b) AR side.

Figure 3.4: Arrow annotations as rendered by both applications.

Comments

Comment annotations are used for the remote expert to convey textual information that would not be easily expressed through the previous annotations, and for the local technician to insert information regarding an inspection element, such as marking it as OK/NOK, add comments and register values. Comment annotations are composed of two elements, visually connected by lines, which can be positioned independently: a small and nonobstructive comment widget, to mark the position to which the comment applies; and a larger comment box, which contains the contents of the comment itself, to be placed where it does not occlude other annotations. Both these elements are defined as billboards, so they are 2D elements that are always pointed at the user's camera. Figure 3.5 shows a comment annotation in both applications. On the VR application, the user can set the content, position and orientation of these annotations, while on the AR application, they cannot be moved and can contain additional controls for maintenance tasks, such as setting the task as OK/NOK (or Not Available (NA), if the equipment is not found), and/or insert text/values, depending on task type. Additionally, for the AR application, to improve the flow of following instructions, after a user sets the status of a task or inserts a text/value, the instructions automatically advance to the next step or, if it is the final step, prompt the user to submit the results of the instructions.

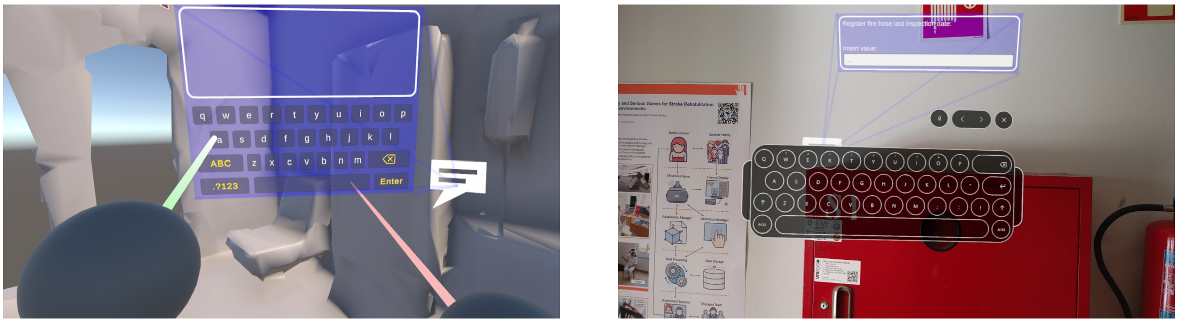

Verify if fire hose, fire extinguisher and fire alarm have visible signaling:
Verify if fire hose, fire extinguisher and fire alarm have visible signaling

(a) VR side.

(b) AR side. Note the 3 additional buttons below the comment box, to mark the task as OK/NOK or NA.

Figure 3.5: Comment annotations as rendered by both applications.

For inserting text and values into the comments, virtual keyboards are used on both applications, as shown in Figure 3.6.



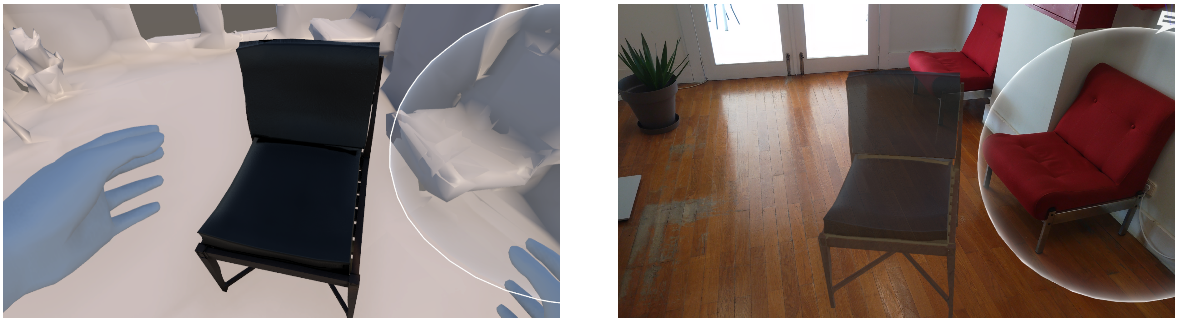
(a) VR side.

(b) AR side.

Figure 3.6: Virtual keyboards for the comment annotations on both applications.

Replicas

Replica annotations are 3D models intended to represent and approximate the look of real objects from the environment. A database of 3D models is required for replica annotations, which should be populated prior to the use of the application, and they can be used for any objects, but common uses would be tools such as wrenches or screwdrivers, to indicate where and how they are to be used, or component parts from equipment to indicate how to assemble/disassemble them or where they are to be moved. Specially useful for this later scenario, replicas also support animations, showing on each step an animation from the current position to the position on the next step. On the AR side they can only be visualized, while on the VR side they can be positioned and rotated, but not scaled, as they are assumed to represent real physical objects with fixed dimensions. Figure 3.7 shows a replica annotation in both applications. It is important to note that while replicas are frequently based on the realistic 3D CAD models of the physical objects that were used for manufacturing them, the prototype, in this instance, utilized models that provided only an approximation of the real object. This was due to the unavailability of realistic 3D CAD models for objects in the local technician's environment used during the evaluation of the prototype, nevertheless, the approximated models were deemed satisfactory for the purpose of showcasing functionality.



(a) VR side. A chair replica is shown at the center, while its physical counter-part was captured as part of the environment, on the right.

(b) AR side. A chair replica is shown at the center, while its physical counter-part is on the right.

Figure 3.7: Replica annotation for a chair as rendered by both applications.

3.3.3 Augmented Reality Application - Local Technician

All user interaction with the application is done through hand gestures, using hand and finger recognition through the HoloLens' cameras. All gestures are performed with a single-hand and are ambidextrous (i.e. either hand can be used to perform them), with the following gestures being recognized:

1. A palm gesture, performed with an open hand, palm facing the user, which opens the hand menu (Figure 3.8).

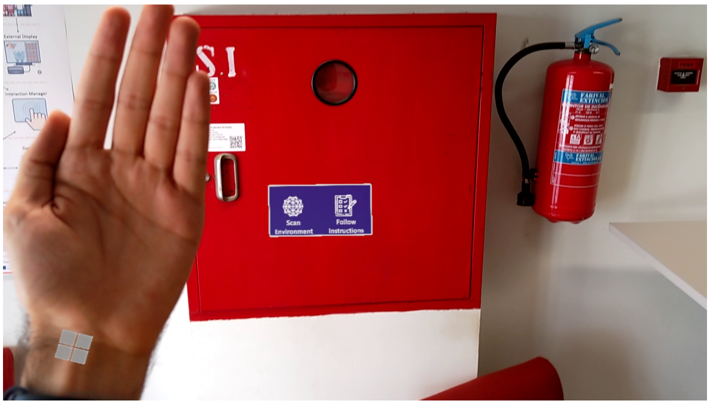


Figure 3.8: Palm gesture being used to open a hand menu, showing the main menu of the application, from which submenus for the environment scan and instruction execution are accessed.

2. A poke gesture, performed using the index finger to intersect or "touch" virtual objects, which is used to activate the interface elements in the hand menu (Figure 3.9), comment annotations (Figure 3.5b).) and virtual keyboard (Figure 3.6b).).

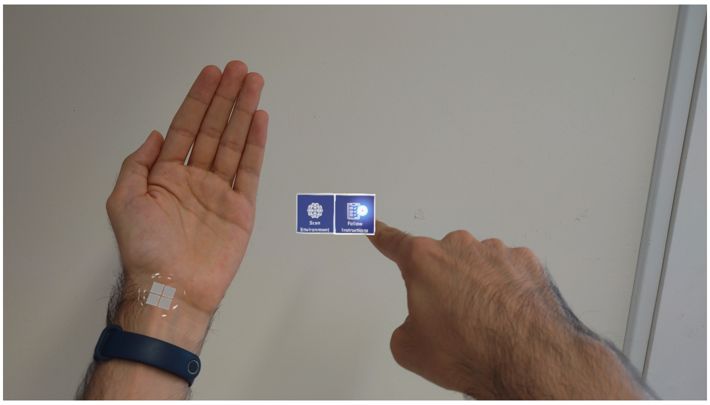
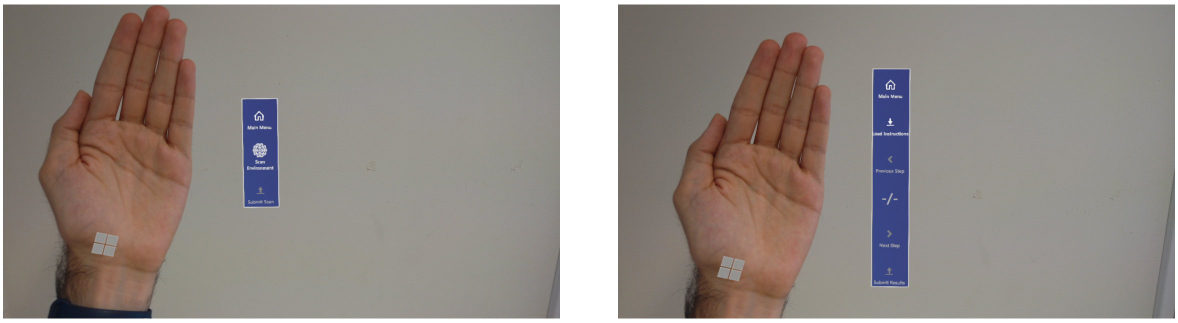


Figure 3.9: Poke gesture being used to interact with interface elements in the hand menu, after the hand menu was opened with the palm gesture.

3. A pinch gesture, which allows using the functionality of the poke gesture from afar. When the user's open hand is visible, palm facing away from the camera, a line is extended from the palm which the user can use to intersect with can object and then "pinch" by joining the thumb and index finger, which will result in the same effect as a poke gesture on the point of intersection.

The hand menu is the main User Interface (UI) for the application. Only essential functions are exposed to make it as intuitive as possible and not overwhelm users, so it is used only to control the state of the application and all configuration options are defined in a separate text file, which is edited outside the application (check Appendix A - System Configuration Files for more detail).

Figure 3.8 shows the main menu, from which submenus for the environment scan (Figure 3.10a) and instruction execution (Figure 3.10b) can be accessed. The interface elements are intuitive and captioned, with buttons to start an environment scan, submit it, load instructions, go forward and backward in instruction steps, show the current step and total number of steps and, finally, submit results.



(a) Submenu for environment scan. The "Submit Scan" button is grayed-out because no environment scan has been initiated yet.

(b) Submenu for instruction execution. The "-/-" on the third row from the bottom keeps tracks of the steps of the loaded instructions, displaying the "current step / total steps". It is currently empty because no instructions are loaded, which is also the reason some of the buttons are grayed-out.

Figure 3.10: Submenus of the AR application.

The 3D reconstruction of the environment makes use of the Microsoft HoloLens 2, without the need for an additional depth sensing device. This exceeds the initial functional requirements which suggested the use of another device for this purpose (section 3.2.2), and greatly simplified the system's usage, and will be discussed in more detail in section 3.4.

After an environment scan is started, the user starts seeing an AR mesh of the 3D reconstruction built so far using depth sensor data, superimposed on the real world. Initially this mesh covers only the area the user is looking at, but it is updated with new data at pre-configured intervals (usually 1s), so the user needs to walk and look around until the mesh covers all relevant areas of his environment to his satisfaction, as illustrated in Figure 3.11. The level of detail of the mesh is adjustable, as is the update interval, through the application's configuration file, though high levels of detail or fast update rates should be considered carefully to not penalize performance.

As described in the section 3.3.1, for the environment to be submitted or for loading instructions, a QR code needs to already have been scanned by the application. For the user to scan a QR code he only needs to look at it, and it will be recognized if it is well in view with good lighting, and marked with a virtual object to confirm recognition. All QR codes are recognized by the application, but only a QR code with the appropriate content is recognized as a usable reference. QR codes with the string content "RemoteAssistanceOrigin" are marked

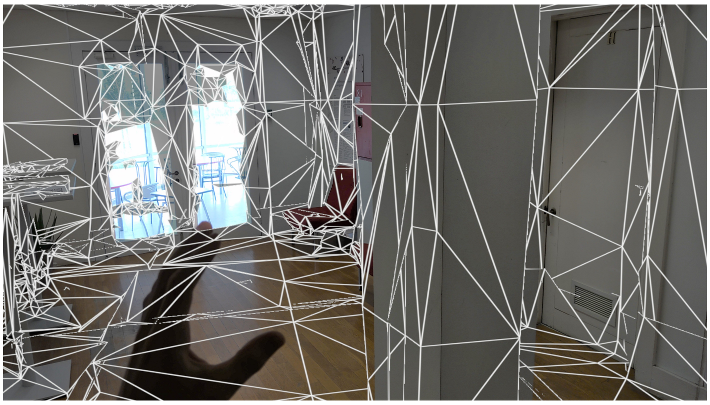
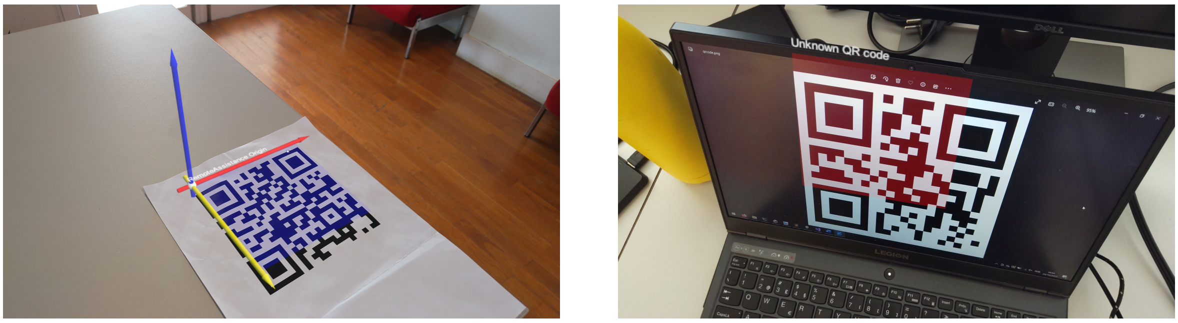


Figure 3.11: Environment being scanned and the AR mesh of the resulting 3D reconstruction.

in green, along with a caption with the QR code's content and axes that illustrate the local coordinate system that will be used as a reference, as shown in Figure 3.12a. All other invalid QR codes are simply marked in red and the caption "Invalid QR code", as shown in Figure 3.12b.



(a) Valid QR code.

(b) Invalid QR code.

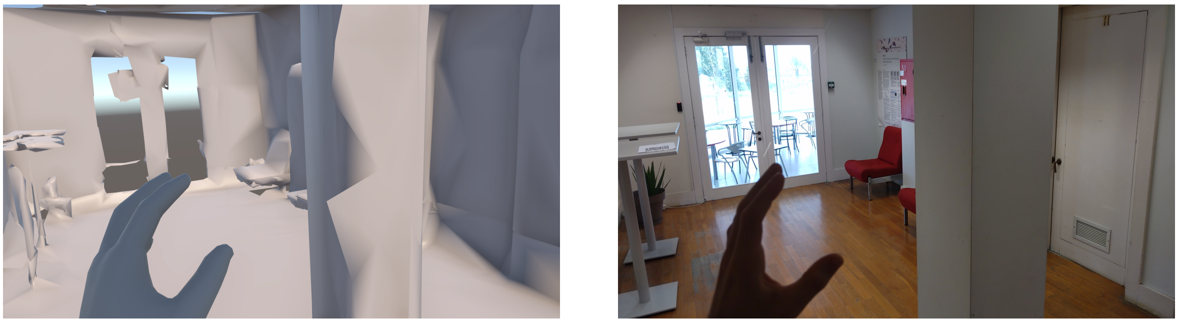
Figure 3.12: QR codes being recognized by the application.

In order for the application to be intuitive and provide a good UX, the UI state machine was devised in a way that makes it easy for the user to understand and follow the intended workflow, with submenus and disabling buttons where appropriate (Figure 3.10). So it is not possible to load instructions while an environment scan is already in progress, and vice-versa, to separate the functionality for the 1$^{\text{st}}$ and 3$^{\text{rd}}$ stages of the workflow described in section 3.3.1. Additionally, if the user attempts to start an environment scan or load instructions before a QR code is loaded, he is also reminded to scan a QR code first.

3.3.4 Virtual Reality Application - Remote Expert

When the application is started, the user is immediately positioned within the 3D reconstruction of the local technician's environment, in the same position and orientation that the technician was when he submitted it, to make the environment recognizable and see it from the technician's perspective. All of this is done transparently from the user's perspective, which exceeds the initial functional requirements (section 3.2.2), as there is no need for any configuration on the VR application to guarantee a proper alignment of the virtual elements in both applications. The details of how this was achieved will be discussed in section 3.4.

The 3D reconstruction is untextured, as this is a limitation of the functions natively available in the HoloLens' SDK where the environment was captured. Figure 3.13 shows a sample of the VE and virtual hands as rendered in the VR application, compared to the original environment.



(a) Virtual environment and virtual hands as rendered in the VR application.

(b) Same scene for the original physical environment.

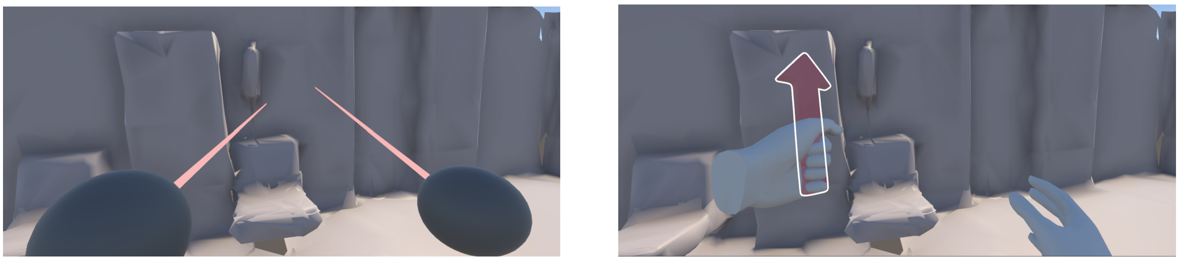
Figure 3.13: Comparison of the virtual environment with the original physical environment.

The expert interacts with the application through the controllers. Locomotion is done through the trackpads, with the left controller moving the user horizontally relatively to where he is looking (forward/backward and strafing), and the right controller rotating the user left/right and moving him vertically, so that taller areas are accessible and the application is viable in places such as staircases. Alternatively, natural locomotion is also supported, by walking and looking around, so the application supports a seated/standing experience, or a room-scale experience, which is useful to adapt to the space requirements where the remote expert will work, and also to manage cybersickness for individuals that are specially susceptible, for which natural locomotion methods are recommended.

The control of the application is modal, with a button press (Menu Button of the Vive controller) to change between two interactions modes:

1. UI interaction mode (Figure 3.14a) - with interactions based on rays that extend from the controllers. Entering/leaving this mode opens/closes the main UI, and when in this mode it is only possible to interact with interface elements, comment annotations and the virtual keyboard.
2. World interaction mode (Figure 3.14b) - in which direct grab interactions are used. It is not possible to interact with interface elements while in this mode and all annotations can be positioned and rotated with grab interactions, while resizable annotations can be scaled with two-handed gestures, by grabbing with each hand and varying the distance between them.

All interactions are ambidextrous and can be performed with either hand. In addition, whenever a virtual object is interacted with it is highlighted with an outline, with audio feedback (in UI interaction mode) or haptic feedback (in world interaction mode) for confirming interactions, as well as grab animations for the virtual hands.



(a) UI interaction mode with ray interactions. On this mode, the virtual hands are replaced by spheres, and rays extend from them, which can only be used to interact with the UI.

(b) World interaction mode with direct grab interactions. On this mode, animated virtual hands are displayed, which can be used to grab virtual objects such as arrows and other annotations but not with the UI.

Figure 3.14: Interaction modes in the VR application.

As for the main UI, it followed the same approach as for the AR application, with a simplified UI with only what is essential to control the state of the application, and any configuration options are defined in a configuration file (check Appendix A - System Configuration Files for more detail). Figure 3.15 shows the main UI, it contains elements for going forward and backward in instruction steps, showing the current step and total number of steps, adding a new step to the instructions and adding each type of annotation. In addition, for adding replica annotations, a replica library is opened below the main UI, as shown in Figure 3.16. In line with good VR design guidelines, a conscious attempt was made to keep the UI small, containing only icons for each button and only showing more information about each in tooltips, so that it does not occlude the environment and annotations, which combined with the fact that it is shown only when invoked should also reduce eye-strain and cybersickness in VR.

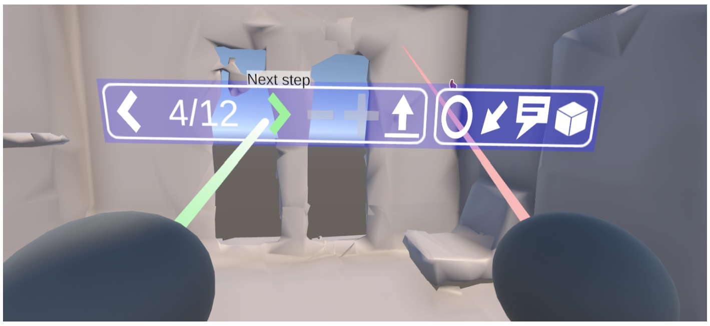


Figure 3.15: Main UI for the VR application, with one of the tooltips.

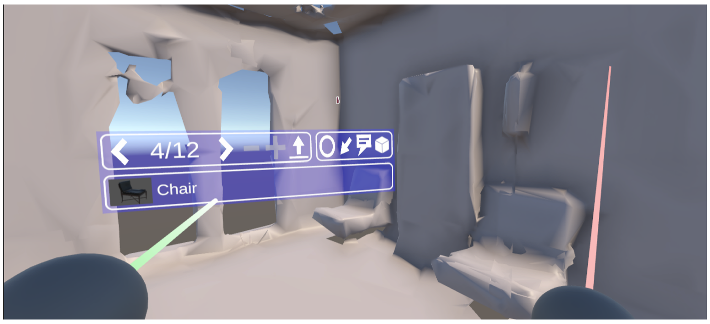


Figure 3.16: Replica library UI for the VR application, attached below the main UI, containing a chair replica.

3.4 Architecture and Implementation

After the description of the system's features and usage, the architecture and the most relevant implementation details will now be discussed.

3.4.1 Spatial Referencing Strategy

Having a consistent spatial reference between the two applications and between remote assistance sessions is one of the main challenges of the implementation. The core of the problem lies in two fronts:

1. The two applications do not share the same spatial coordinates origin, as they run in distinct devices in different locations.
2. The HoloLens device used for the AR application is an untethered device and, as such, does not have a fixed frame of reference. As a result, the coordinates' origin will not be located in the same physical position between two use sessions on the device.

The initially proposed strategy for tackling the problem was to use a QR code as a common frame of reference between the two applications, with a physical QR code on the AR side, and a virtual QR code on the VR side positioned by the user during a configuration step on an equivalent position on the VE's 3D reconstruction. In addition, it was also proposed that the 3D reconstruction would be generated by an additional device.

Early in the AR implementation of the prototype, after recognizing the HoloLens' depth sensing capabilities, it was decided that they would be used for creating the 3D reconstruction instead of an external device. This not only made the system more convenient to use, it also opened up new possibilities to approach the problem of spatial referencing.

Most game engines allow objects to be organized in a tree hierarchy, for which each object has a local transformation to position and orient it relatively to the parent object and the final transformation is a concatenation of all transformations of the objects all the way to the root. To illustrate the implemented solution, taking advantage of this transformation hierarchy, Figure 3.17 shows the hierarchy for a single annotation on each stage of the system workflow.

In the 1$^{\text{st}}$ stage, the HoloLens aligns the environment model with the physical environment, so the transformation, relative to the QR code, that aligns the model with the physical environment, is computed and stored. Next, on the 2$^{\text{nd}}$ stage, all annotations are referenced

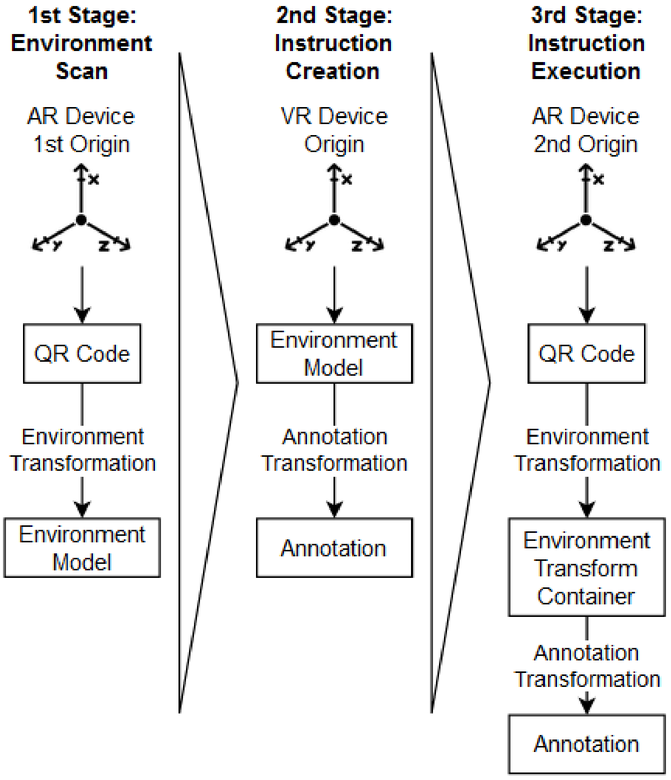


Figure 3.17: Transformation hierarchy for an annotation on each stage of the system workflow.

relative to the loaded environment model, and these are the transformations that are stored on the file that defines the instructions. Finally, on the 3rd stage, the QR code is again used as a parent and the environment transformation that was stored in the 1st stage is applied before the annotation transformation stored in the instructions file for each annotation. As the annotations were referenced with respect to the environment model and the transformation stored in the 1st stage is the one that aligns the environment model with the physical environment, this ensures the annotations align with the physical environment as well. Since the annotations on the VR application were referenced relative to the environment model, there is no need to consider any transformation or for any configuration on this stage, as the absolute origin of the VR device is irrelevant. Likewise, since the QR code object was always used as a reference in the AR device, the absolute origin of the AR device is not significant either.

3.4.2 Architecture

Figure 3.18 shows an architecture diagram of the system with the most relevant runtime entities and a focus on the flow of data during the system workflow, omitting the details concerning the interactions in AR/VR.

Besides the two applications, the existence of a common storage element is also implied. However, this is not necessarily the case, both applications allow configuring the path that is used to store and load files, so while this can be e.g. a network location on a server, which would work as shown on the diagram, it can also be a location on the AR/VR device

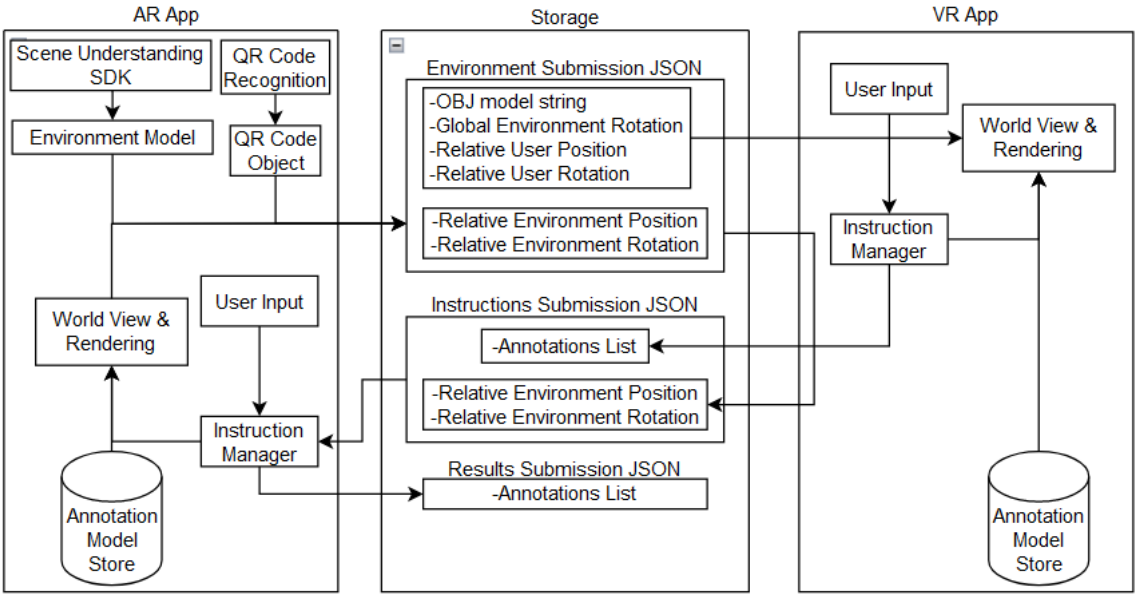


Figure 3.18: Architecture diagram for the system prototype.

itself. In the later case, it would require transferring files between the stages of the system workflow, but since the applications work asynchronously and are not running simultaneously, from a functional standpoint it is equivalent to using a common storage and is a valid simplification for the sake of clarity. Likewise, for exchanging data with the JSON files, each file has an associated runtime entity that encapsulates all information and has functionality to serialize/deserialize itself to/from a file, as it would not make sense to incur the performance penalty of frequent storage accesses at runtime, but these implementation details were also omitted.

Following the system workflow, the process starts on the AR side with scanning a QR code (making use of some functionality in the HoloLens[4], with some tweaks) and is used to create a virtual object at the QR code's location. Afterward, the environment is scanned, using the Scene Understanding SDK, which is responsible for gathering the depth sensor data from the HoloLens for a point cloud, and then using it to generate a 3D model of the environment aligned with the physical world. When the user starts the environment submission, the following information is computed and saved to a JSON file:

- Relative Environment Position and Relative Environment Rotation - the transformation parameters that align the environment model with the physical world relative to the QR code.
- OBJ model string - the environment model in Wavefront OBJ format, which, as a plaintext format, can easily be saved as a string field in the JSON file.
- Global Environment Rotation - the rotation transformation parameters that set the orientation of the environment model relatively to the global origin. This is required because the generated environment model is not aligned with the user's point of view

[4] https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview

through its vertex information, so this information is used to make the environment recognizable and easy to navigate on the VR side.
- Relative User Position and Relative User Rotation - the transformation parameters that define the position of the AR user, i.e. the application's camera or world view, relative to the environment model. This is used to place the VR user in the VE in the same equivalent position and rotation as the AR user in the physical environment when he initiated the submission.

Next, the instructions are authored on the VR application. The Instruction Manager in both applications is responsible for managing the annotations in a set of instructions, adding and editing annotations, organizing them into steps and controlling which annotations are rendered. Both applications have the same annotations but the particularities of each device and SDK require that they are adapted for rendering and interaction on each device, therefore each application's Annotation Model Store contains models with slight variations. This, combined with the fact that instructions need to be portable between applications, necessitated that instructions are defined using an abstraction where each annotation is defined through an encapsulating class, easily serializable as text, with an ID for the type of annotation it represents and all other necessary information such as its position, rotation, OK/NOK tag and comment text (if applicable). This is the format in which instructions are defined in the Annotations List that is placed in the JSON file when the VR user submits his instructions, along with the Relative Environment Position and Relative Environment Rotation, which is unused and only passed to the application to make sure it is available for visualizing the instructions on the AR side.

Finally, on the last stage of the process on the AR application, the instructions are loaded and rendered by the Instruction Manager, as already described, with relevant edits such as setting OK/NOK tags based on user input and finally saving the results to a JSON file in the format described for the Annotations List.

More detail about all files involved in the process and the fields of the encapsulating class for annotations can be found in Appendix B - Environment, Instructions and Results Files.

3.5 Tools used

Unity 2021.3[5]

Unity is one of the most popular game engines today and the main development tool for the prototype. Game engines have in many ways become the standard for development of VR and AR applications[6], and Unity has good support for the targeted devices of the prototype, is very accessible and has a large community for support and contributions to its Asset Store, so Unity is a good choice for the prototype. Its scripting language is C#, so all the implementation was made in this programming language.

[5] https://unity.com/

[6] *Global Game Engines Market Report 2020-2027: Growing Trend of AR/VR and High Demand for Game Engines for Gamification Applications*, https://www.globenewswire.com/en/news-release/2020/12/17/2147205/28124/en/Global-Game-Engines-Market-Report-2020-2027-Growing-Trend-of-AR-VR-and-High-Demand-for-Game-Engines-for-Gamification-Applications.html Accessed: 2023-06-21

OpenXR[7]

OpenXR is an open standard for an API for communicating with eXtended Reality (XR) devices. It is meant to make applications more cross-platform by allowing developers to target this open API instead of a distinct proprietary API for each device. Choosing OpenXR was relevant for the VR application, as initially it was suggested that it could target both the HTC Vive and the Meta Quest 2. The Meta Quest 2 ended up not being tested, but minimal work should be required to support it, since as a result of choosing the OpenXR standard, the application became more platform-independent and can easily be made to work both with other XR devices with OpenXR support, or, for development, other desktop operating systems with available OpenXR runtimes and build export options in Unity.

Mixed Reality Toolkit (MRTK) 2.8[8]

MRTK is the native SDK for development of AR applications in the HoloLens, so it was absolutely required and used for the AR application. It provides functionality for all interactions, using the devices' sensors and the building blocks required for most applications.

XR Interaction Toolkit[9]

The XR Interaction Toolkit is a framework for VR and AR development, which is available as a Unity package and slightly lower-level than MRTK. All the interactions for the VR application made use of its components, with some customizations such as for two-handed gestures, which required subclassing an existing component.

Scene Understanding SDK[10]

The Scene Understanding SDK allows using the sensor data from a Windows Mixed Reality device such as the HoloLens and creating representations of the environment. It integrates with the MRTK, which also provides features for using sensor data, but are no longer in development and are being moved to this new SDK (though the MRTK is still in active development for other features). So, the Scene Understanding SDK was chosen for this functionality to provide a more solid experience and use new features such as being able to scan a larger environment.

Other repositories and assets from the Unity Asset Store:

- QR code tracking in Unity[11]: A sample project to demonstrate how to use QR codes in the HoloLens, whose code was used with some adjustments to work on more recent versions of Unity.
- Json.NET Converters[12]: An Asset Store package to allow the native JSON serialization classes to work easily with Unity classes.

[7] https://www.khronos.org/OpenXR/

[8] https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05

[9] https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html

[10] https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/scene-understanding-sdk

[11] https://github.com/microsoft/MixedReality-QRCode-Sample

[12] https://assetstore.unity.com/packages/tools/input-management/json-net-converters-simple-compatible-solution-58621

- Occulus hand models[13]: Since the XR Interaction Toolkit used on the VR application does not include hand models with grab animations and no free, ready to use, toolkit for this purpose was found either, these models were used, for which the skeletal animations were created and configured.
- Quick Outline[14]: For highlighting interactable objects in VR.
- Responsive Keyboard[15]: Smartphone virtual keyboard asset, which was adapted for use in VR.
- Runtime OBJ Importer[16]: Used to convert OBJ models to Unity objects.
- Scene OBJ Exporter[17]: Used to convert an Unity object's model to OBJ. Used for the environment scan after adapting it to work at runtime (previously only worked within the Unity editor).
- Settings / Config file[18]: To use configuration files for both applications.
- Patio Chair[19]: For use as a replica annotation in the user tests.
- Icon packs for use in the UI:
  - UX Flat Icons[20].
  - Clean Vector Icons[21].
  - Minimal UI Sounds[22].
  - OSVR GUI Framework[23].

[13] `https://developer.oculus.com/downloads/package/oculus-hand-models/`

[14] `https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488`

[15] `https://assetstore.unity.com/packages/tools/input-management/responsive-keyboard-mobile-games-customisable-214794`

[16] `https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547`

[17] `https://assetstore.unity.com/packages/tools/utilities/scene-obj-exporter-22250`

[18] `https://assetstore.unity.com/packages/tools/settings-config-file-81722`

[19] `https://assetstore.unity.com/packages/3d/props/furniture/patio-chair-222957`

[20] `https://assetstore.unity.com/packages/2d/gui/icons/ux-flat-icons-free-202525`

[21] `https://assetstore.unity.com/packages/2d/gui/icons/clean-vector-icons-132084`

[22] `https://assetstore.unity.com/packages/p/minimal-ui-sounds-78266`

[23] `https://assetstore.unity.com/packages/tools/gui/osvr-gui-framework-145482`

CHAPTER 4

Prototype Evaluation

In this chapter, the experimental setup for the prototype evaluation through a user study will be described, followed by the results and their discussion.

4.1 Experimental Setup

The research conducted for this dissertation took place within the context of a research group (VARLab[1]), which had previous evaluation work on instruction authoring. Consequently, it was deemed more relevant to the group to focus on the side of instruction execution for evaluation of this dissertation's prototype, and only the 3$^{\text{rd}}$ stage of the system workflow presented in section 3.3.1) was evaluated.

The experimental setup involved two conditions:

- Condition 1 (C1): Traditional method making use of a paper with a checklist of the tasks, which the user annotates and then inserts the result in a laptop, to simulate the usual workflow of making the results of a WO available on a CMMS.
- Condition 2 (C2): AR method using a HoloLens 2 and a QR code printed on a piece of paper for spatial referencing.

The usability tests took place in a small section of an office building's corridor (Figure 4.1), with both natural and artificial illumination. This section was chosen because it contains multiple items that are relevant for an inspection round in a small area, well within the effective range of the application, which targets room-scale experiences. The aforementioned relevant items are:

- Exit door;
- Access card reader;
- Evacuation button;
- Evacuation plan;
- Fire extinguisher;

[1] `https://sites.google.com/view/varlab/home`

- Fire hose;
- Fire alarm;
- Fire extinguisher, fire hose and fire alarm signalling;
- Chair (obstruction) in front of the fire extinguisher;
- Corridor lighting;
- Insect killer trap box.

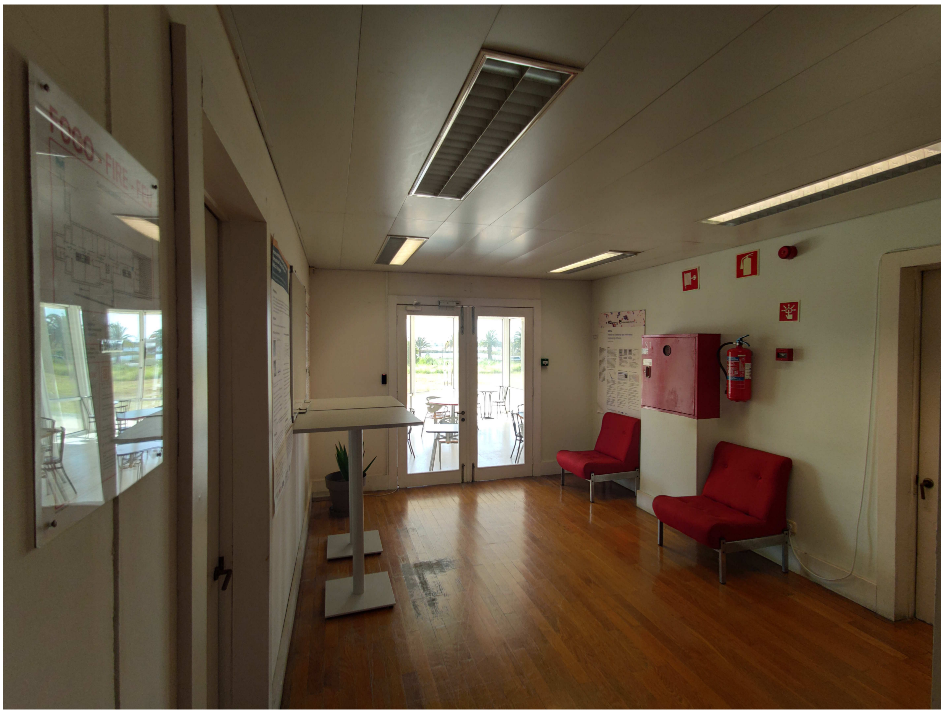


Figure 4.1: Location of the user tests.

4.2 Experimental Design

The experimental design followed a within-groups approach, with a single independent variable: the experimental condition (C1 or C2). As such, all participants went through both conditions sequentially, in a randomized order (ensuring balance between conditions), in an attempt to account for practice effects, with a null hypothesis that both conditions were equally usable and acceptable for the proposed tasks. As for dependent variables, they were the participants' impressions and performance metrics, while secondary variables were demographic data and familiarity with AR.

4.3 Tasks

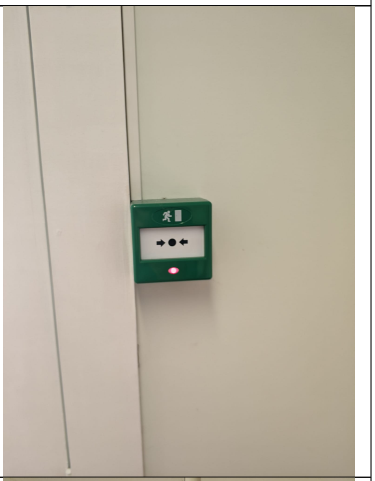
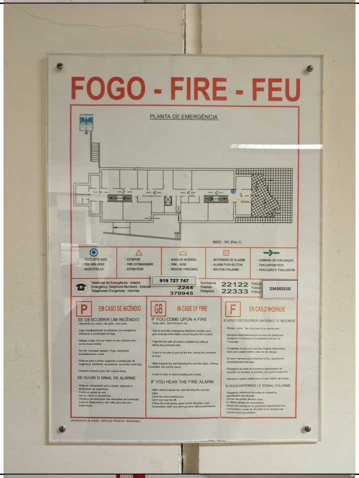
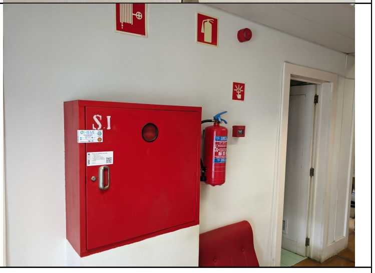
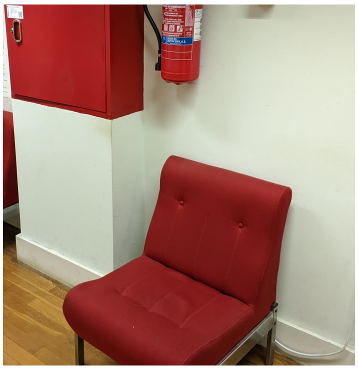
The assigned tasks were the same for both conditions, and were intended to approximate real preventive maintenance tasks for an inspection round of a building, making use of the existing equipments in this section of the building.

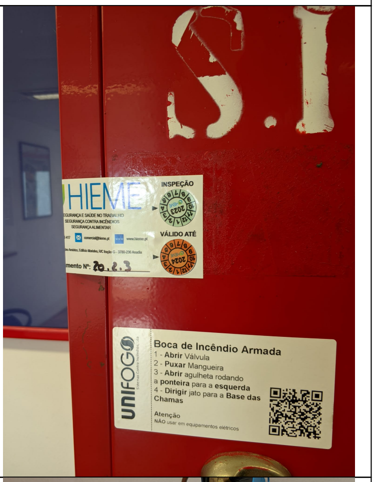
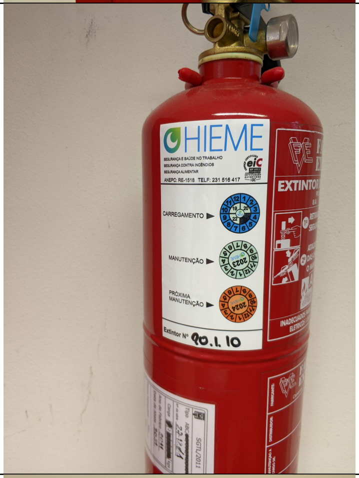
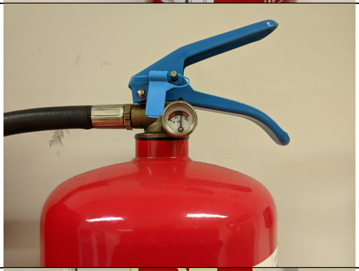
The tasks were of 3 possible types, following the specifications defined for the application scenario (Section 3.1):

- OK/NOK task - a task where the equipment needs to be classified with an "OK" or "Not OK" tag. In addition, though it is not standard for a WO, the "NA" tag was also added for the purpose of the tests, for when the user fails to find or identify the equipment.
- OK/NOK task, with comment if NOK - similar to the previous task type but, if NOK, also requires that the user inserts a comment to indicate why the equipment is not NOK. To make the comments less subjective for the purpose of the tests, it was only applied to the lighting, and the users were instructed to merely report how many elements are not operational.
- Register value task - the user registers a value from an equipment, such as the last inspection date.

Table 4.1 shows the tasks performed in the user study, while the form used for condition C1 can be found in Appendix D - Inspection Round Task Results Questionnaire.

Nr.	Task Description	Task Type	Picture
1	Verify if exit door is locked.	OK/NOK	
2	Verify if access card reader is turned on.	OK/NOK	

Nr.	Task Description	Task Type	Picture
3	Verify if exit button is operational (red LED turned on).	OK/NOK	
4	Verify if evacuation plan is visible.	OK/NOK	
5	Verify if fire hose, fire extinguisher and fire alarm have visible signalling.	OK/NOK	
6	Move obstructions such as chairs from the front of the fire extinguisher to somewhere where it is nonobstructive, such as the other side of the corridor.	OK/NOK	

Nr.	Task Description	Task Type	Picture
7	Register fire hose last inspection date.	Register value	
8	Register fire extinguisher last maintenance date.	Register value	
9	Register fire extinguisher pressure.	Register value	
10	Verify that fire alarm glass is unbroken.	OK/NOK	

Nr.	Task Description	Task Type	Picture
11	Verify if all lights are operational.	OK/NOK, comment if NOK	
12	Verify that insect killer trap box is not full.	OK/NOK	

Table 4.1: List of tasks defined for the user tests, in the order they were executed.

4.4 Measurements

Two main types of data were collected during the experiment:

- **Participants' impressions**, gathered through questionnaires and a small interview at the end of the experiment. The full questionnaire can be found in Appendix C - User Study Post-experience Questionnaire but the general outline is:
  - 6 main dimensions to assess how the 2 conditions compare in the participants' opinion, in the form of 6 questions targeting both conditions, with a 7 point Likert scale, rated from low to high. The following dimensions were defined:
    - D1 - Level of attentional allocation;
    - D2 - Effectiveness in perceived information understanding;
    - D3 - Level of confusion or distraction about the content used;
    - D4 - Level of physical effort;
    - D5 - Level of mental effort;
    - D6 - Level of satisfaction.
  - A standard System Usability Scale (SUS) questionnaire for the system;
  - A few additional questions regarding the system, including open-ended questions for additional user commentary.
- **Performance metrics**, namely:
  - Time to perform each task, measured in seconds.

- Whether each task was successful, by comparing the output of each task (i.e. the OK/NOK tags assigned, commentary or registered values, if applicable) with the correct result.

4.5 Procedure

Before starting the user study, both the system and the building section were prepared:

- For the C1 condition, a laptop was placed on a table close to the user for inserting values.
- For the C2 condition, the spatial referencing QR code was affixed on a nearby pillar and then, using the system, the environment was scanned and the instructions created. Using the annotations of the system, there are many ways in which they can be combined to express a sequence of tasks. Some care was taken to be consistent for the whole set of instructions, making sure that each type of annotation had a single and well-defined use:
  - Area of interest - used to identify the equipment(s) of the current task.
  - Comment - used for describing the current task and insertion of its status or value.
  - Arrow - used exclusively to point to the comment annotation that defines the next task after the user finishes his current task.
  - Replica - used for tasks that require moving objects.

  Note that without textures, there is some difficulty in aligning the annotations with the intended environment features in VR, still, even though this could be improved with trial-and-error and can impact the results, it was felt that this would be a more accurate representation of the current state of the system.

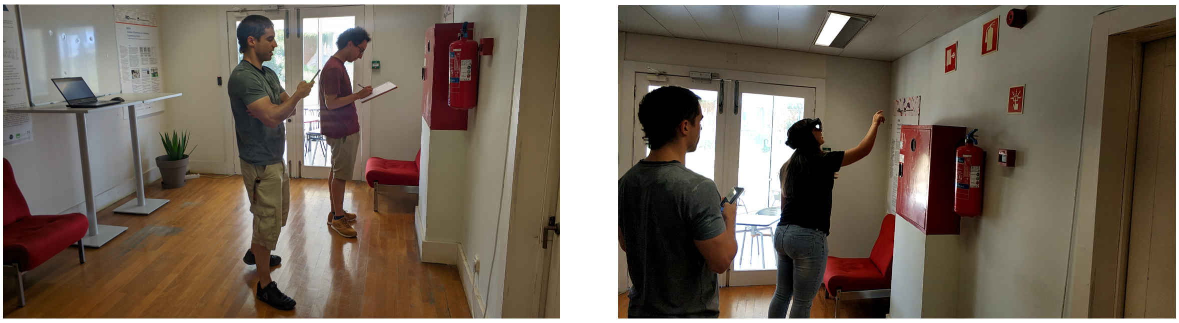
Afterward, the following procedure was followed for each participant:

1. The participant agrees to participate in the experiment, explicitly accepting the conditions in the informed consent.
2. The user is given an overview of the system and its purpose, and briefed on the types of tasks to be performed, as well as how to read the dates on the equipments[2].
3. Out of view of the participant, the chair is placed in front of the fire extinguisher, as an obstruction for task nr. 6.
4. The user is prepared to follow the instructions, which depends on the condition for the test run he will execute:
   - For condition C1: the user is given the printed WO and a pen.
   - For condition C2: the user is given the HoloLens 2 device with the application, it is calibrated for his eyes, and then, in a distinct area from where the test run will take place, he is given an overview of the system's interactions and UI, followed by an adaptation period with a small set of demonstration instructions, for which he is provided assistance whenever necessary (Figure 4.2).

[2] Since the date format on the fire extinguisher and fire hose is not intuitive, this was a precaution so that it does not affect the experiment. An illustrative photo from the date on another fire extinguisher was used as demonstration, without disclosing what equipment it belonged to.

Figure 4.2: User study participant during the adaptation period with the system, being provided an overview of the system's usage and assistance whenever necessary.

5. The user follows the instructions, while he is observed and the completion time of each task is measured. For condition C2, the result of each task is stored in the system automatically, while for condition C1, the user inserts the result in the laptop at the end of each task.



(a) Condition 1 (C1).

(b) Condition 2 (C2).

Figure 4.3: User study participants following each condition.

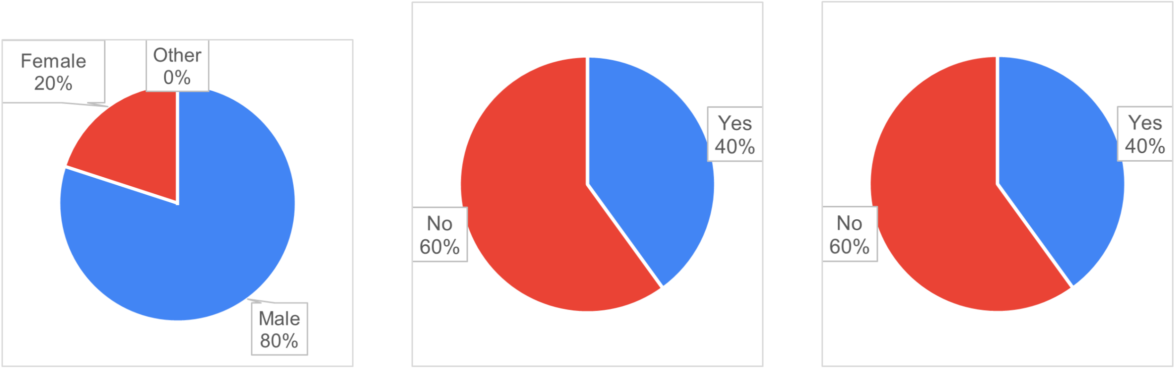
6. If the user has not completed both conditions yet, the process goes back to step 4., now following the remaining condition.
7. After completing the test runs for both conditions, the user answers the post-study questionnaire, followed by a small interview to gather any other further insights and observations.

The entire procedure took around 30mins for each participant.

Demonstration videos of the environment scan[3] and instruction authoring[4] can be found on YouTube, as well as footage from an inspection run for the HoloLens condition with one of the participants[5].

4.6 Participants

A total of 10 participants were recruited for the user study, of which 80% were male and 20% female, with a mean age of 23.9 (SD=2.8). The recruited participants were all students in multiple fields, mostly at the master's level, with a few in bachelor or doctorate degrees. Finally, 40% of participants had previous experience in AR, and also 40% of participants had previous experience in VR. In both cases, in mostly recreative scenarios such as games and some from participation in user tests for other dissertations in the field.



(a) Gender. **(b)** Experience with AR. **(c)** Experience with VR.

Figure 4.4: Distribution of participants by gender and previous experience with AR/VR.

4.7 Results and Discussion

Both the objective and subjective results of the evaluation will now be presented and discussed.

4.7.1 Performance Metrics

First, a breakdown of the tasks in the defined inspection round will be provided to account for the influence of different types of tasks. Figure 4.5 shows the average across all participants for the time taken to perform each task. Naturally, the tasks that require inserting text require more time, namely tasks 7 to 9 and task 11. Generally, the system performed worse than the traditional method for such tasks and better for simple OK/NOK tasks (tasks 1-5, 10 and 12). This is consistent with the initial expectations, since using the virtual keyboard for inserting text is not as efficient as using a pencil or a physical keyboard, which was further confirmed by the user feedback from the survey and interviews. Task 11 was an OK/NOK task that required a comment if NOK, so in addition to requiring the use of the virtual keyboard it also

[3] `https://youtu.be/5UZwTV-tcjs`
[4] `https://youtu.be/6dv7iKQnATE`
[5] `https://youtu.be/MJM-nN5BHPc`

required a visual inspection, and as the objects to inspect were placed along a wide area it was also penalized by the reduced Field of View (FOV) imposed by the headset, according to user feedback. Task 6 required moving an object, which accounts for the longer time to perform it and took longer to perform for the HoloLens due to the participant taking some time to follow the animation of the object with his gaze first before performing it, with some hesitation due to the aforementioned reduced FOV while using the headset.

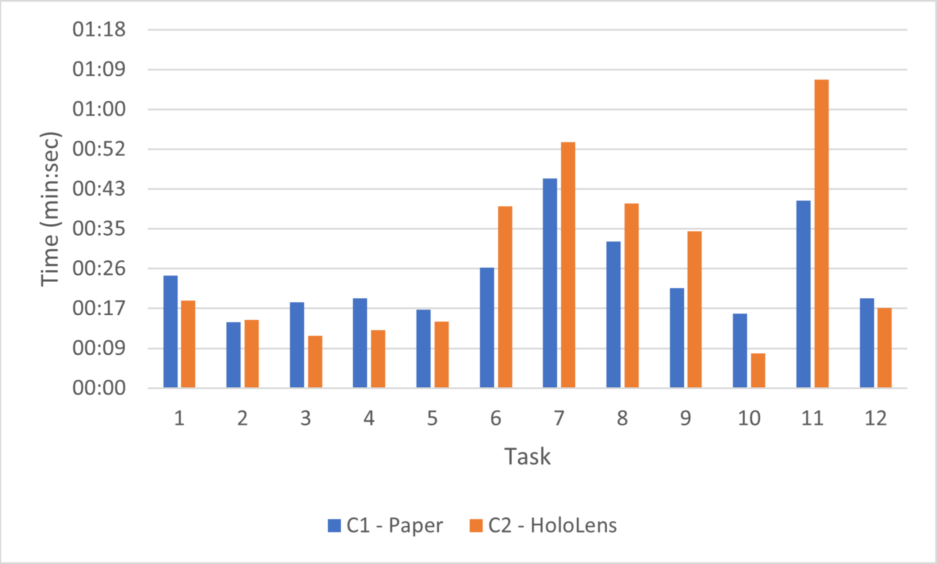


Figure 4.5: Average task time for each task.

Figure 4.6 shows the number of inspection rounds with a failed task for each of the tasks. The criteria for defining the success of each task were whether the equipment was correctly classified as OK/NOK, the inserted values were correct, or the objects to be moved were placed on the correct position. Most failures occurred on the tasks that required inserting text, which is also consistent with the times in Figure 4.5. From the text inserted by the users on these tasks, it appears that the HoloLens helped with locating the correct labels on the equipment that contained the values to insert, but this effect was largely offset by typing errors on the virtual keyboard, which accounts for the higher number of failures on such tasks on some cases. For all other tasks the system performed much better, with no failures whatsoever, as the equipments are easier to locate with virtual objects.

Figure 4.7 shows the average time to complete all tasks from an inspection round. To account for learning effects, averages were evaluated first considering all inspection rounds, then considering only the inspection rounds that were the first performed by each participant and finally only the ones that were performed afterward. Considering all inspection rounds the system performed worse than the traditional method, mostly due to the tasks that required using the virtual keyboard, as the total time was shorter without those tasks, which shows the system is an efficient alternative to the traditional method for OK/NOK tasks that do not use the virtual keyboard. Without learning effects, the system performed better than the traditional method, as expected, since finding and identifying the equipments was the main hurdle and virtual objects are much more efficient than text to convey spatial information to

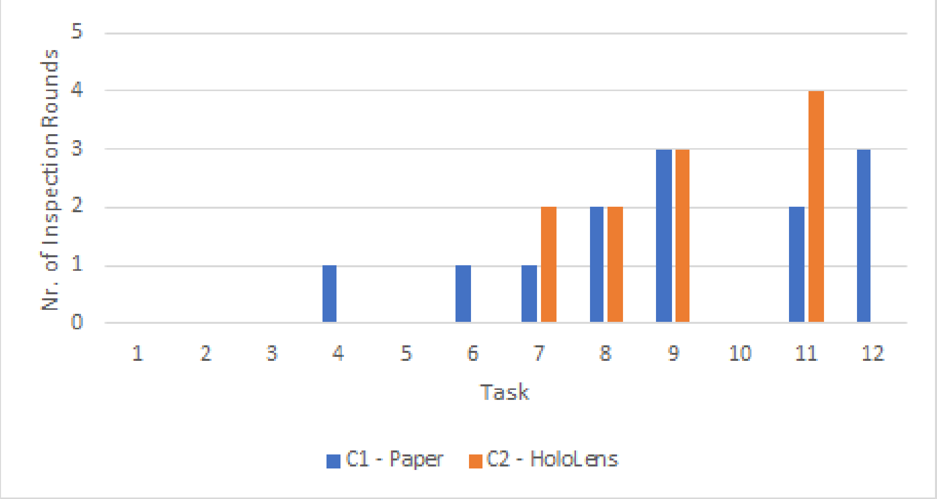


Figure 4.6: Inspection rounds with failures at each task.

assist with finding an object or understanding where to move it. This is further emphasized when comparing the average time for the paper condition with and without learning effects, which had a huge impact on the total time, reducing it by 45.1%, while it remained relatively constant for the system. Such a reduction showcases the potential for the system to be used as a training tool for new technicians still unfamiliar with a particular inspection round, while still retaining better performance on the user's first round.

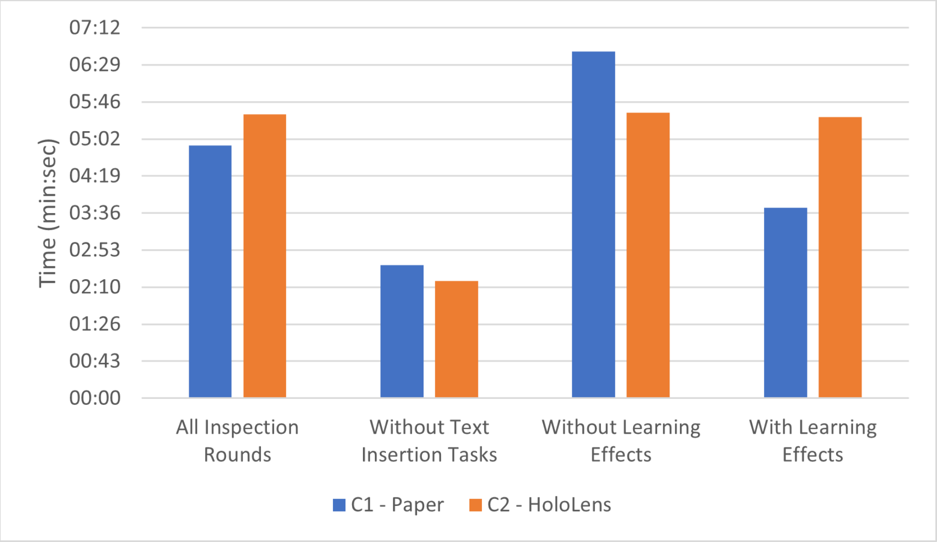


Figure 4.7: Average time to complete all tasks from an inspection round.

Figure 4.8 shows the average failure rate for the tasks on each inspection round. The system performed better than the traditional method, likely due to the aforementioned advantages in locating the equipments. Since most of the task failures when using the system seem to have been due to mistypes on the virtual keyboard, removing the tasks that required text insertion results in a 100% success rate for the system, while the traditional method still shows some failures due to not finding the equipments. Once again, a single inspection round prior to using the traditional method had a very significant impact on its performance, with a reduction of 80% on its failure rate.

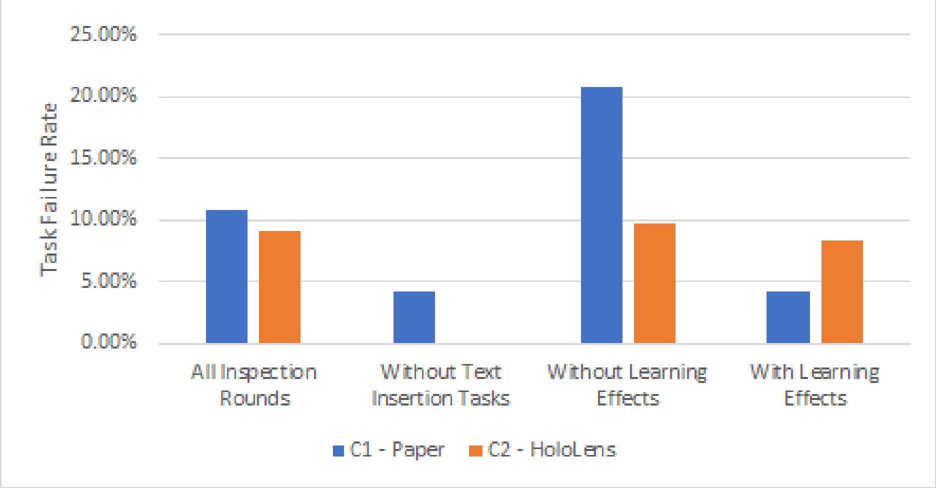


Figure 4.8: Average failure rate for all tasks from an inspection round.

4.7.2 User Impressions

The usability of the AR application was evaluated in the post-experience survey using a standard SUS questionnaire, which resulted in a SUS score of 75.3, which is in the upper half for a system with good usability, above the standardized value of 68 for good usability and below 80.3, which would indicate excellent usability.

Figure 4.9 shows a comparison between both conditions according to user impressions from other questions. The participants felt that the system required significantly higher levels of attentional allocation (dimension D1) than the traditional method, which is to be expected given that they were using an unfamiliar system and a considerable part of them (40%) were even new to AR technology in general. However, despite the additional attentional requirements for the system, the users did not feel that the content used was confusing or distractive (dimension D3) and actually felt that, when compared to the traditional method, the effectiveness in understanding perceived information (dimension D2) was higher and the level of mental effort (dimension D5) was similar. This shows that, in the view of the users, the virtual content was not obstructive and effectively augmented their environment with relevant information. In particular, the fact that users scored the content of the traditional method as actually more confusing/distracting, showcases the advantage of using a AR headset for such tasks, as having to look away from the environment to read and write values in the paper or laptop is distractive. As for the level of physical effort (dimension D4), it shows that the hand interactions with virtual content were comfortable to use, as it scored very low for the system, while the traditional method on the other hand required some physical effort, as it required the user to carry the notepad and pen, walk around in search of the equipments and to move toward the laptop.

As shown in Figure 4.10, users showed significantly more preference for the C2 condition (HoloLens) compared to the traditional method, which is also consistent with the results

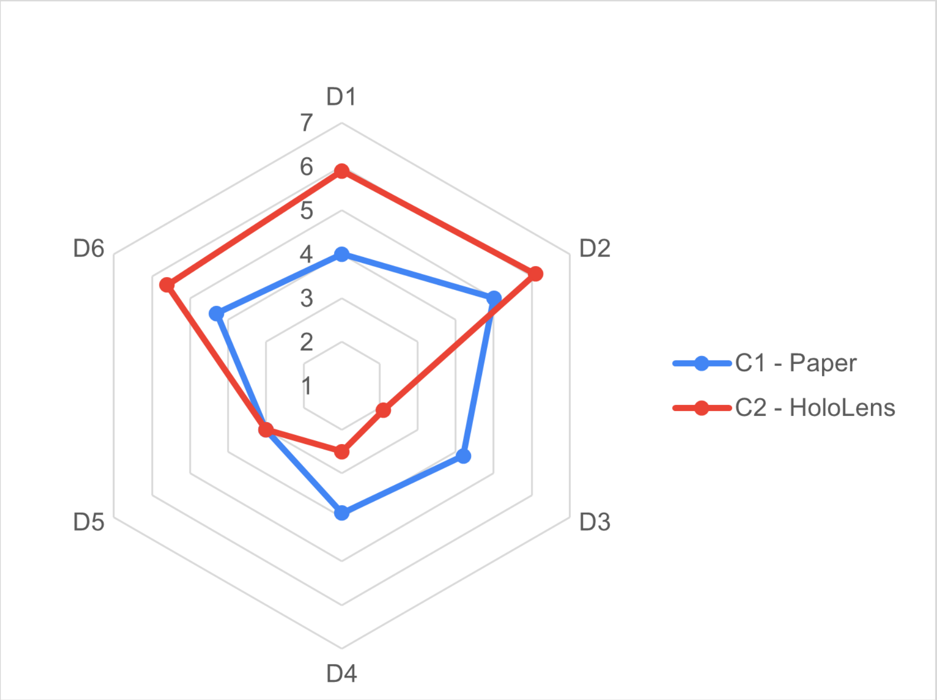


Figure 4.9: Radar chart representing a comparison of both conditions according to user impressions, with the following dimensions: D1 - Level of attentional allocation; D2 - Effectiveness in perceived information understanding; D3 - Level of confusion or distraction about the content used; D4 - Level of physical effort; D5 - Level of mental effort; D6 - Level of satisfaction. Values are an average for all participants for a Likert scale: 1 - Low; 7 - High.

for dimension D6 of Figure 4.9. While preference by itself does not necessarily guarantee increased performance for all tasks, it is important to keep workers motivated, which can indirectly affect performance and is, perhaps, even more important for training scenarios. In addition, it is a good subjective measure of both the system usability and how receptive they would be to use the system daily at their jobs.

Users most commonly reported that the main advantage of the system compared to the traditional method is that it made it easier to find the equipments. The fact that it did not require recording on both the paper and the laptop was also frequently mentioned. Other mentioned advantages included reduced distractions, enhanced task comprehension, making it easier to find the equipments allowed them to focus on more complex problems, and the added benefit of hands-free operation. On the other hand, one of the users noted the imperfect alignment of some annotations, which resulted from the lack of textures during authoring as mentioned in section 4.5). Since such comments were not common among the users and the performance metrics showed promising results, it is likely that these alignment issues did not have a very significant impact on the results, as though they might impact the annotations that require more precision (such as locating the last inspection date label on a fire extinguisher), those labels were still relatively easy to find and the main hurdle is finding the equipments themselves, in which the annotations are still effective even with small misalignments.

As for additional features to improve the system, users mostly suggested ways to make

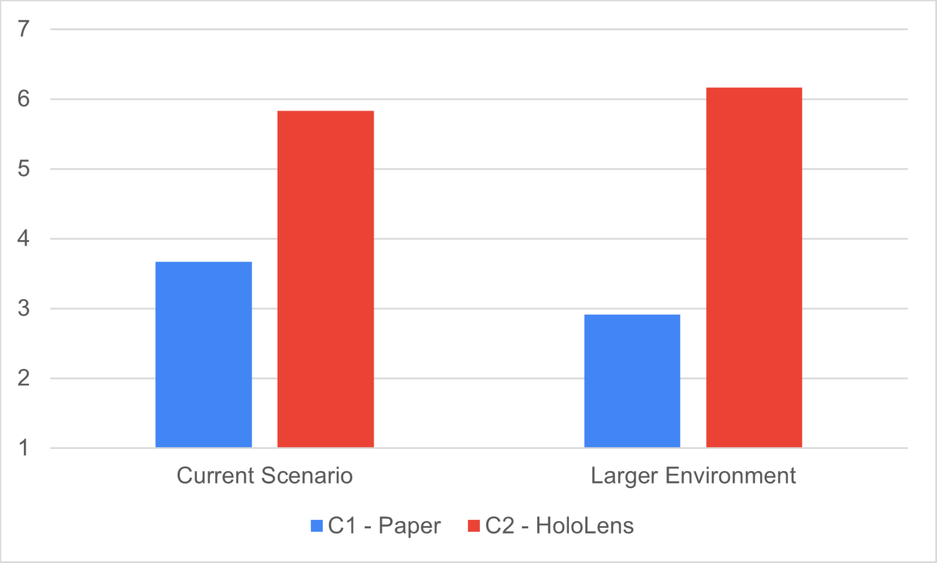


Figure 4.10: Preference of the participants between both conditions, averaged for all users in a 7-point Likert scale: 1 - Low; 7 - High.

it even easier to find the next task. Keeping the arrow visible and always pointing to the next task until their gaze is directed toward it was the most commonly mentioned. Other suggestions included tracing a path toward the next task, using 3D audio to help locate it, and replacing the text instructions by voice instructions which would also help the user navigate the environment.

The assigned tasks were generally very simple to execute, as evidenced by the user reactions to statement 5 in Table 4.2. Furthermore, for a new recruit, most of the difficulty in an inspection round stems from locating and identifying the equipments, which is of course harder in a larger area. It is likely that for more complex tasks and a larger area, the system would have better results when compared to the traditional method. The user preferences in Figure 4.10 seem to support this notion, as the difference in preferences between both conditions increased in favour of the system and harder tasks would likely result in a larger difference.

The remaining user feedback in Table 4.2 will now be considered. The users felt that the interactions with the virtual objects were intuitive (statement 1), which is consistent with the result of the SUS score. The fact users felt that their performance with the system would improve with training time is promising (statement 2), as users are used to traditional methods in their daily lives, so a comparable level of experience with the system could make it significantly more competitive in comparison.

80% of the users felt that the system had potential to be used for remote collaboration scenarios, while 20% were not sure. Furthermore, looking at statement 3 and 4 in Table 4.2, users also felt that the system could be used for assembly/disassembly tasks, and that the existing annotations were enough for any task. This shows the flexibility of the system and its potential for its application in other scenarios.

Statement	**Likert Score**
1) The interaction with the virtual content is intuitive.	6
2) The performance for completing tasks with this system will improve with training time.	7
3) I think the system could be used for more complex instructions such as assembly/disassembly of industrial equipment or other equipment that requires specialized knowledge.	6
4) The existing virtual objects are enough to help with any task.	6.5
5) The assigned tasks were too difficult.	1.5

Table 4.2: User reactions to various statements about the system in a Likert scale (1 - Strongly Disagree; 7 - Strongly Agree), with a median for all users.

User feedback from free text questions on the survey and interviews, as well as from observation during the user study, resulted in some further insights:

- The fact that all annotations in the instructions had a clear and well-defined role, with no extraneous annotations, likely had a very positive impact on the results. For example, even though they were briefed on the types of annotations used by the system, many users did not regard the arrow annotation as an annotation authored by a remote expert but as just a built-in feature of the system, so it is likely that using the arrow for any other purpose would create confusion and distraction. This shows the importance of good authoring for the instructions, that leads the user through a consistent workflow.
- As described in section 3.3.3, the AR application allows users to interact with the comment annotation, in which they would insert the task result, through either the poke gesture (a direct touch interaction) or the pinch gesture (without direct contact), so both were demonstrated during the adaptation period. Even though the pinch gesture is intended to be used to interact with objects when they are far away and is not as efficient when they are close, as the user often needs to take a step back, some users opted to use it almost exclusively. Though situations such as this would likely not be so common with more training in the system, it again shows the importance of having a clear workflow and not overwhelming the user with options.
- The HoloLens slightly reduces the user's FOV so when defining a task it is important to attempt to place all annotations so that they can be visible simultaneously within the FOV of the user. Task 6 and task 11 (Table 4.1) would likely have benefitted from better placement following this principle.

CHAPTER 5

Conclusions and Future Work

5.1 Conclusions

In this dissertation, a framework for remote assistance was successfully defined along with a reference implementation, with support for view independence through 3D model reconstruction, asynchronous assistance through step-by-step instructions and virtual replica annotations. The system is unique in the literature in combining all of these characteristics, specially in regard to the focus on asynchronous assistance. Moreover, the system allows a high degree of flexibility, as it can be used in any location, encompassing on its workflow the environment scan of each new location, the authoring of the instructions in immersive VR and execution of the instructions in AR, with no additional devices besides the HMDs for each user. All of this is accomplished with minimal user configuration for spatial referencing, as only the placement of a physical QR code on the local user's environment is required. Note also that both the authored instructions and results are defined in plain text JSON files, so they can be easily read and edited by external tools, increasing the system's flexibility and interoperability.

The prototype was then evaluated for the most relevant stage of the system workflow: instruction execution. The user study focused on a maintenance inspection round use case, comparing the traditional method (Condition 1 - Paper) with the system (Condition 2 - HoloLens), while gathering performance metrics and the impressions of the participants through surveys and interviews.

The system showed better performance than the traditional method for task failure rate but worse for total time. Nonetheless, without text insertion tasks, the system still performed better for total time, as inserting text through the virtual keyboard was its main drawback, but was still efficient in helping find the equipments to inspect, which is the main difficulty for an inspection round. For the traditional method, users that ran an inspection round using the system beforehand had a 45.1% reduction in total time and 80% reduction in task failure rate compared to the ones that did not, while for the HoloLens, using the traditional method first had almost no impact on their metrics. This presents strong evidence of the system's potential as a training tool for this particular use case.

Users showed preference towards the system compared to the traditional method, which is a positive indication of the system's usability, how receptive they would be to adopt it and can also have an impact on performance by keeping them motivated. The system's good usability was further confirmed by the results of the SUS questionnaire, with a score of 75.3, on the upper half of the range of standardized scores that indicate good usability. User impressions indicate that the virtual content was not distractive and was effective in conveying relevant information and actually reduced confusion compared to the traditional method, as users did not need to look away from the equipments to register values. Lower levels of physical effort were also reported for the system, as the users did not need to walk as much and carry a notepad, while still retaining similar levels of mental effort. Users also mentioned the benefit of hands-free operation and being able to focus on more complex problems.

Users also pointed-out some ways in which the system could be improved. One of the main reported drawbacks in usability was inserting text through the virtual keyboard, for which either storing audio communication, or speech-to-text/text-to-speech solutions were suggested as possible alternatives.

The system is limited to room-scale experiences, and the tasks of the user study were quite simple, where the main hurdle was simply finding the equipment. Since in a larger area, equipments would be much harder to find, it is likely that the results would significantly shift further in favor of the system. Users seemed to recognize this, as this is what happened when questioned regarding their preference between both methods if they were applied to a larger area. In addition, one of the suggestions was to improve the way the local user is directed to the next task after finishing his current task, which would become even more relevant in a larger area. Some possible suggested solutions were keeping an arrow always visible and pointing to the next task until their gaze is directed toward it, another would be tracing a path to the next task, which could be useful if the user needs to navigate between walls and equipment.

Users also felt that their performance with the system would improve with training time. This is promising, as given their familiarity with conventional methods, a similar level of experience with the system could enhance its competitiveness.

Finally, the users generally agreed that the system could be used for other tasks, such as assembly/disassembly of equipments, and that the existing annotations were sufficient for any task. Though the system was not evaluated for such scenarios, both these impressions and the positive objective results with the current scenario are a good indication of the system's flexibility and it potentially not being limited to the evaluated use case only as a viable solution.

5.2 Future Work

Though the initial requirements for the system were exceeded in some regards, some further directions for future work were identified, which were not pursued due to time constraints and either not being directly related or essential to the goals this dissertation originally set out to accomplish.

The main improvement to the prototype that was identified would be assistance for larger areas. It would allow it to be used in a much wider variety of situations, such as part picking in a warehouse, or locating equipment in a large factory before it is serviced. The MRTK toolkit contains some functionality to accomplish this, such as Spatial Anchors[1] and World Locking Tools (WLT) [2] . WLT, in particular, would allow the recognition of previous physical environments, which could even allow the prototype to not require any QR code for referencing, by re-using the origin initially set on the first run in a physical environment for referencing instead.

Enabling the local technician to add his own annotations when the environment is scanned (1st stage of the system workflow) to be seen by the remote expert, could also greatly improve communication, by allowing the local technician to better explain the problem with verbal and non-verbal cues. This could be specially relevant if combined with some degree of synchronous assistance.

Synchronous assistance, is in fact another major feature that could prove very convenient for situations where either the environment cannot be assumed to be static or the instructions' complexity requires closer supervision. Semi-synchronous assistance is also an interesting possibility, where the users can control when their views and annotations are synchronized. Views could be synchronized after the environment undergoes a relevant change, such as after opening a panel in a machine that is receiving maintenance, while instructions could be shared in batches, so that while the local user completes a batch, the expert is working on the next.

The prototype could also be made substantially more flexible by allowing the use of external models for the 3D reconstruction. This could also address the aforementioned problem of having textured geometry, as some devices support creating textured models, including finer meshes and camera detail for the textures that can exceed the HoloLens' capabilities.

Additional work on the prototype's evaluation could also be considered. Performing a user study which includes other stages of the system workflow would be the most obvious one. Moreover, if a user study for the complete user workflow was performed, considering multiple cycles of the workflow could be another possibility. This could be a way to simulate semi-synchronous assistance, but with the remote expert always waiting for the local technician to complete the previous batch of instructions and submit a new environment scan. This would make it possible to assess the potential time-savings of semi-synchronous assistance, without requiring additional features to be implemented for the prototype. Finally, studying other use cases to evaluate the system's flexibility for other scenarios would also be interesting, particularly for assembly/disassembly tasks, which are a good fit for the use of replica annotations with animations, as they illustrate the positioning of the parts very well.

[1] *Spatial anchors - Mixed Reality | Microsoft Learn*, `https://learn.microsoft.com/en-us/windows/mixed-reality/design/spatial-anchors` Accessed: 2023-07-03

[2] *World Locking Tools documentation | Microsoft Learn*, `https://learn.microsoft.com/en-us/mixed-reality/world-locking-tools/` Accessed: 2023-07-03

Bibliography

[1] R. Druta, C. Druta, P. Negirla, and I. Silea, «A Review on Methods and Systems for Remote Collaboration», *Applied Sciences*, vol. 11, no. 21, p. 10 035, 2021.

[2] B. Marques, S. Silva, R. Maio, L. Vale Costa, P. Dias, and B. Sousa Santos, «Remote Work Is Here to Stay! Reflecting on the Emerging Benefits of Mixed Reality Solutions in Industry», in *International Conference on Human-Computer Interaction*, Springer, 2023, pp. 253–260.

[3] M. Rice, S. C. Chia, H. H. Tay, *et al.*, «Exploring the Use of Visual Annotations in a Remote Assistance Platform», in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016, pp. 1295–1300.

[4] B. Ens, J. Lanir, A. Tang, *et al.*, «Revisiting Collaboration Through Mixed Reality: the Evolution of Groupware», *International Journal of Human-Computer Studies*, vol. 131, pp. 81–98, 2019.

[5] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva, «The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature», *Frontiers in psychology*, p. 2086, 2018.

[6] P. Wang, X. Bai, M. Billinghurst, *et al.*, «AR/MR Remote Collaboration on Physical Tasks: A Review», *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102 071, 2021.

[7] M. Sereno, X. Wang, L. Besançon, M. J. McGuffin, and T. Isenberg, «Collaborative Work in Augmented Reality: A Survey», *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2530–2549, 2022.

[8] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock, «Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction», in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–14.

[9] S. Kathleen, S. Sven, N. B. Claudia, and E. Frank, «Fulfilling Remote Collaboration Needs for New Work», *Procedia Computer Science*, vol. 191, pp. 168–175, 2021.

[10] E. A. Isaacs and J. C. Tang, «What Video Can and Can't Do for Collaboration: A Case Study», in *Proceedings of the first ACM International Conference on Multimedia*, 1993, pp. 199–206.

[11] S. H. Choi, M. Kim, and J. Y. Lee, «Situation-dependent Remote AR Collaborations: Image-based Collaboration Using a 3D Perspective Map and Live Video-based Collaboration with a Synchronized VR Mode», *Computers in Industry*, vol. 101, pp. 51–66, 2018.

[12] J. Heiser, B. Tversky, and M. Silverman, «Sketches For and From Collaboration», *Visual and spatial reasoning in design III*, vol. 3, pp. 69–78, 2004.

[13] D. Anton, G. Kurillo, and R. Bajcsy, «User Experience and Interaction Performance in 2D/3D Telecollaboration», *Future Generation Computer Systems*, vol. 82, pp. 77–88, 2018.

[14] D. Calandra, A. Cannavò, and F. Lamberti, «Improving AR-powered Remote Assistance: A New Approach Aimed to Foster Operator's Autonomy and Optimize the Use of Skilled Resources», *The International Journal of Advanced Manufacturing Technology*, vol. 114, no. 9-10, pp. 3147–3164, 2021.

[15] R. Schroeder, *Possible Worlds: The Social Dynamic of Virtual Reality Technology.* Westview Press, Inc., 1996, p. 25.

[16] S. S. Kardong-Edgren, S. L. Farra, G. Alinier, and H. M. Young, «A Call to Unify Definitions of Virtual Reality», *Clinical Simulation in Nursing*, vol. 31, pp. 28–34, 2019.

[17] M. V. Sanchez-Vives and M. Slater, «From Presence to Consciousness through Virtual Reality», *Nature Reviews Neuroscience*, vol. 6, no. 4, pp. 332–339, 2005.

[18] J. Steuer, F. Biocca, M. R. Levy, *et al.*, «Defining Virtual Reality: Dimensions Determining Telepresence», *Communication in the age of virtual reality*, vol. 33, pp. 37–39, 1995.

[19] H. L. Miller and N. L. Bugnariu, «Level of Immersion in Virtual Environments Impacts the Ability to Assess and Teach Social Skills in Autism Spectrum Disorder», *Cyberpsychology, Behavior, and Social Networking*, vol. 19, no. 4, pp. 246–256, 2016.

[20] J. Q. Coburn, I. Freeman, and J. L. Salmon, «A Review of the Capabilities of Current Low-cost Virtual Reality Technology and Its Potential to Enhance the Design Process», *Journal of computing and Information Science in Engineering*, vol. 17, no. 3, 2017.

[21] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, «Recent Advances in Augmented Reality», *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, p. 34, Nov. 2001.

[22] P. Milgram and F. Kishino, «A Taxonomy of Mixed Reality Visual Displays», *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[23] M. Tait and M. Billinghurst, «The Effect of View Independence in a Collaborative AR System», *Computer Supported Cooperative Work (CSCW)*, vol. 24, no. 6, pp. 563–589, 2015.

[24] J. Lanir, R. Stone, B. Cohen, and P. Gurevich, «Ownership and Control of Point of View in Remote Assistance», in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2243–2252.

[25] O. Oda, C. Elvezio, M. Sukan, S. Feiner, and B. Tversky, «Virtual Replicas for Remote Assistance in Virtual and Augmented Reality», in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 405–415.

[26] B. Marques, S. Silva, A. Teixeira, B. Santos, and P. Dias, «Concepts and Methods to Support the Development and Evaluation of Remote Collaboration Using Augmented Reality», Ph.D. dissertation, Dec. 2021.

[27] P. Wang, Y. Wang, M. Billinghurst, H. Yang, P. Xu, and Y. Li, «BeHere: A VR/SAR Remote Collaboration System Based on Virtual Replicas Sharing Gesture and Avatar in a Procedural Task», *Virtual Reality*, pp. 1–22, 2023.

[28] H. Tian, G. A. Lee, H. Bai, and M. Billinghurst, «Using Virtual Replicas to Improve Mixed Reality Remote Collaboration», *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2785–2795, 2023.

[29] X. Zhang, X. Bai, S. Zhang, *et al.*, «A Novel MR Remote Collaboration System Using 3D Spatial Area Cue and Visual Notification», *Journal of Manufacturing Systems*, vol. 67, pp. 389–409, 2023.

[30] F. Trojan and R. F. Marçal, «Proposal of Maintenance-types Classification to Clarify Maintenance Concepts in Production and Operations Management», *Journal of Business Economics*, vol. 8, no. 7, pp. 560–572, 2017.

[31] K. Khazraei and J. Deuse, «A Strategic Standpoint on Maintenance Taxonomy», *Journal of Facilities Management*, 2011.

[32] P. Bourque, R. E. Fairley, and I. C. Society, *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*, 3rd. Washington, DC, USA: IEEE Computer Society Press, 2014.

[33] F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, and K. Z. Zamli, «Software requirements prioritisation: A systematic literature review on significance, stakeholders, techniques and challenges», *IEEE Access*, vol. 6, pp. 71 497–71 523, 2018.

[34] A. Cooper, *The inmates are running the asylum.* Springer, 1999.

[35] T. Miaskiewicz and K. A. Kozar, «Personas and User-centered Design: How Can Personas Benefit Product Design Processes?», *Design studies*, vol. 32, no. 5, pp. 417–430, 2011.

Appendix A - System Configuration Files

For both applications, the external configuration file is a plain text file in JSON format in the device's storage.

AR Application

```
{
        "debugMode":false,
        "dataPath":"",
        "meshLevelOfDetail":0,
        "meshUpdateInterval":1.0,
        "firstAutoUpdateDelay":0.0
}
```

Code 1: Example of configuration file for the AR application.

Code 1 shows an example of the external configuration file of the AR application. It contains the following fields:

- `debugMode` - a boolean value that activates debug mode. In this mode a previously created environment submission file is loaded, and the 3D reconstruction of that environment is shown with the same pose that would be applied to the annotations when an instructions submission file is loaded, to test the alignment of the 3D reconstruction with the physical environment.
- `dataPath` - a string with the path for the folder where the environment submission and results files will be created and the instructions submission file will be read from. It can be a location in the device or an accessible network location in a server. If left blank, the default will be used, which was set as Unity's persistentDataPath (`https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html`).
- `meshLevelOfDetail` - an integer that defines the level of detail of the mesh of the 3D generated 3D reconstruction of the environment. Valid values are:
  - 0 - coarse;
  - 1 - medium;
  - 2 - fine;
  - 255 - unlimited.

- `meshUpdateInterval` - a float representing the interval in seconds between updates to the 3D reconstructed mesh;
- `firstAutoUpdateDelay` - a float representing the interval in seconds before the first update of the 3D reconstruction after it is first initiated.

VR Application

```
{
    "debugMode":false,
    "dataPath":""
}
```

Code 2: Example of configuration file for the VR application.

Code 2 shows an example of the external configuration file of the VR application. It contains the following fields:

- `debugMode` - a boolean value that activates debug mode. In this mode, a previously created instructions submission file is loaded, and its instructions are displayed for editing and testing their alignment with the virtual environment.
- `dataPath` - a string with the path for the folder where the instructions submission file will be created, and the environment submission file will be read from. It can be a location in the device or an accessible network location in a server. If left blank, the default will be used, which was set as Unity's persistentDataPath (`https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html`).

Appendix B - Environment, Instructions and Results Files

All the files involved in the system's workflow are plain text files in JSON format. They are always created or read using the folder defined in the settings file (as described in Appendix A - System Configuration Files).

The environment submission file always has the name `environment.json`. Code 3 is provided as an example of its contents. Note that while positions are defined using Cartesian coordinates, rotations are defined using quaternions and therefore 4 dimensions, which is true for all other files as well.

The instructions submission and results submission files `instructions.json` and `instructions.json`, respectively. They share the same format, with the same fields, their only difference is that the results file contains more information in the fields for task results, such as OK/NOK tags and inserted values and comments. Code 4 is provided as an example of their contents.

The `steps` field contains the instructions themselves, defined as an array of arrays of JSON annotation objects, with the first index identifying the step of the instructions while the second index selects among all annotations in that step. Each JSON annotation object results from an annotation class defined for both applications, containing the following fields:

- `prefabName` - a string that essentially identifies the annotation type, using the name of the corresponding Unity prefab. Replica annotations depend on a shared model database, so there can be an arbitrary number of such annotations, each with their own prefab and unique name, but for other annotations types, the following prefab names are always recognized:
  - `VolumeOfInterest` - Volume of interest annotation;
  - `Arrow` - arrow annotation;
  - `CommentWidget` - the comment widget of a comment annotation;
  - `CommentBox` - the comment box of a comment annotation.
- `localPosition` - relative position of the annotation, in Cartesian coordinates.
- `localRotation` - relative rotation of the annotation, expressed as a quaternion.
- `localScale` - relative scale of the annotation, used for resizing.
- `tracked` - a boolean indicating whether the annotation is tracked. A tracked annotation indicates that it is related to other annotations in the previous or next step, such as

```
{
  "objString": "OBJFormatString"
  "relativeEnvironmentPosition": {
    "x": -0.9206556,
    "y": -0.0178837776,
    "z": 0.296335965
  },
  "relativeEnvironmentRotation": {
    "x": -0.585022,
    "y": 0.200912267,
    "z": 0.7636447,
    "w": 0.1850141
  },
  "globalEnvironmentPosition": {
    "x": -0.441476822,
    "y": -0.0604096949,
    "z": 0.6264245
  },
  "globalEnvironmentRotation": {
    "x": 0.5173667,
    "y": -0.6829095,
    "z": -0.409606636,
    "w": 0.313350677
  },
  "relativeUserPosition": {
    "x": -0.246746257,
    "y": -0.8241845,
    "z": -0.109949559
  },
  "relativeUserForward": {
    "x": 0.109203637,
    "y": 0.7122067,
    "z": -0.6934234
  }
}
```

Code 3: Example of an environment submission file. Note that `"OBJFormatString"` in the `objString` field is merely a placeholder for a string in OBJ format that defines the 3D reconstruction of the environment, which was replaced for clarity, as it is very long.

when it is animated between its current and next position. Also, for the VR application, a tracked annotation is automatically placed on the next step on its previous position, when creating a new step, so that the user can set the position of the next related annotation, such as when positioning parts in an assembly process.

- `previousStepAnnotationIndex` - an integer that for tracked annotations indicates the index of its related annotation in the previous step. If it is not a tracked annotation then it is set as -1.
- `nextStepAnnotationIndex` - an integer that for tracked annotations indicates the index of its related annotation in the next step. If it is not a tracked annotation then it is set as -1.
- `annotationText` - a string with the text content of the annotation, if applicable, such as in comment annotations.
- `commentType` - a string that defines the type of comment, only applicable to comment annotations. The following strings are valid:

```json
{
  "relativeEnvironmentPosition": {
    "x": -0.9206556,
    "y": -0.0178837776,
    "z": 0.296335965
  },
  "relativeEnvironmentRotation": {
    "x": -0.585022,
    "y": 0.200912267,
    "z": 0.7636447,
    "w": 0.1850141
  },
  "steps": [
    [
      {
        "prefabName": "VolumeOfInterest",
        "localPosition": {
          "x": -0.157811716,
          "y": -0.765476,
          "z": -0.9931822
        },
        "localRotation": {
          "x": 4.47034836E-08,
          "y": -4.47034836E-08,
          "z": 0.0,
          "w": 1.0
        },
        "localScale": {
          "x": 0.133790016,
          "y": 0.13379,
          "z": 0.133790016
        },
        "tracked": false,
        "previousStepAnnotationIndex": -1,
        "nextStepAnnotationIndex": -1,
        "annotationText": null,
        "commentType": "Simple",
        "okNok": -1,
        "localComment": "",
        "value": ""
      }
    ]
  ]
}
```

Code 4: Example of an instructions or results submission file, with a single annotation in the first step of the instructions, for simplicity.

- `Simple` - a simple comment to provide more information, with not additional features;
- `OkNok` - a comment to describe a task that (for the AR application) has buttons to indicate if the task is OK, NOK or NA.
- `Value` - a comment to describe a task that (for the AR application) has a field to insert a value.
- `CommentIfNok` - similar to the "OkNok" comment type, but if the NOK option is chosen in the AR application then a field is also shown for the user to insert a

comment detailing why it is NOK.

- `okNok` - an integer used to indicate the state of the task, for the comment types `OkNok` and `CommentIfNok`. The following values are valid:
  - -1 - for NA.
  - 0 - for NOK.
  - 1 - for OK.
- `localComment` - a string containing the comment inserted by the user on the AR application for comments of `CommentIfNok` type.
- `value` - a float containing the value inserted by the user on the AR application for comments of `Value` type.

Appendix C - User Study Post-experience Questionnaire

This user study consists in the use of an application with Augmented Reality capabilities for completing a sequence of tasks. The resulting data will be used for research purposes and to help advance an MSc Dissertation, by understanding the application's benefits and limitations when compared to traditional solutions. All provided information is entirely confidential and will not be distributed or used for any purpose other than this research. As a participant in this study, I declare that I am aware that I will participate in an experience that will be augmented with virtual content, and that this experience could be filmed or photographed. I declare as well that I am aware that all collected data in this study will be used for scientific ends only, guaranteeing the anonymity of all participants. I understand that, at any point, I am free to remove my consent or refuse participation in this study. In the event of any questions or problems concerning my participation, I will contact: Ivo Félix, MsC student (ifelix@ua.pt); Paulo Dias, PhD (paulo.dias@ua.pt); Bernardo Marques, PhD (bernardo.marques@ua.pt).

□ I have read and agree to the treatment of my personal data in the terms described above.

- Which condition did you try first?
  □ Condition C1 - Paper
  □ Condition C2 - AR Tool

Condition C1 - Paper

All information provided is entirely confidential and will not be distributed or used for any purpose other than this research.

Thank you for your collaboration.

1. D1- Level of attentional allocation:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

2. D2- Effectiveness in perceived information understanding:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

3. D3- Level of confusion or distraction about the content used:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

4. D4- Level of physical effort:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

5. D5- Level of mental effort:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

6. D6- Level of satisfaction:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

Condition C2 - AR Tool

All information provided is entirely confidential and will not be distributed or used for any purpose other than this research.

Thank you for your collaboration.

1. D1- Level of attentional allocation:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

2. D2- Effectiveness in perceived information understanding:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

3. D3- Level of confusion or distraction about the content used:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

4. D4- Level of physical effort:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

5. D5- Level of mental effort:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

6. D6- Level of satisfaction:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

7. I think that I would like to use this system frequently.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

8. I found the system unnecessarily complex.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

9. I thought the system was easy to use.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

10. I think that I would need the support of a technical person to be able to use this system.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

11. I found the various functions in this system were well integrated.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

12. I thought there was too much inconsistency in this system.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

13. I would imagine that most people would learn to use this system very quickly.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

14. I found the system very cumbersome to use.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

15. I felt very confident using the system.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

16. I needed to learn a lot of things before I could get going with this system.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

17. The interaction with the virtual content is intuitive.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

18. The performance for completing tasks with this system will improve with training time.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

19. The existing virtual objects are enough to help with any task.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

20. I think the system could be used for more complex instructions such as assembly/disassembly of industrial equipment or other equipment that requires specialized knowledge.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

21. What were your main difficulties when using the system?

22. What features do you feel were missing from the system?

23. Add any other additional observations that you feel are relevant about the system.

    ____

    ____

    ____

Overall Ratings

All information provided is entirely confidential and will not be distributed or used for any purpose other than this research.

Thank you for your collaboration.

1. Rate your preference towards Condition C1 - Paper for the scenario used:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

2. Rate your preference towards Condition C2 - AR Tool for the scenario used:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

3. Rate your preference towards Condition C1 - Paper Tool for a larger environment:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

4. Rate your preference towards Condition C2 - AR Tool for a larger environment:

	1	2	3	4	5	6	7	
Low	○	○	○	○	○	○	○	High

5. The assigned tasks were too difficult.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

6. Do you think Condition C2 - AR Tool has the potential to be used for scenarios of remote collaboration? (e.g., receiving instructions from a remote expert)?
   □ Yes.
   □ No.
   □ Not sure.
7. What were the main benefits of Condition C2 - AR Tool compared to Condition C1 - Paper Tool?

    ____

    ____

    ____

8. Leave other comments you may find useful.

____

____

____

Demographic Survey

All information provided is entirely confidential and will not be distributed or used for any purpose other than this research.

Thank you for your collaboration.

1. Age: ____
2. Gender:

   □ Male.

   □ Female.

   □ Other.
3. Occupation: ____
4. Have you ever used VR applications?

   □ Yes.

   □ No.

   □ Not sure.
5. If yes, in what contexts? ____
6. Have you ever used AR applications?

   □ Yes.

   □ No.

   □ Not sure.
7. If yes, in what contexts? ____

Appendix D - Inspection Round Task Results Questionnaire

Work Order

Inspection round tasks.

1. Verify if exit door is locked:

   □ OK

   □ NOK

   □ NA
2. Verify if access card reader is turned on:

   □ OK

   □ NOK

   □ NA
3. Verify if exit button is operational (red LED turned on):

   □ OK

   □ NOK

   □ NA
4. Verify if evacuation plan is visible:

   □ OK

   □ NOK

   □ NA
5. Verify if fire hose, fire extinguisher and fire alarm have visible signaling:

   □ OK

   □ NOK

   □ NA
6. Move obstructions such as chairs from the front of the fire extinguisher to somewhere where it is nonobstructive such as the other side of the corridor.

   □ OK

   □ NOK

   □ NA
7. Register fire hose last inspection date: ______________________________
8. Register fire extinguisher last maintenance date: ________________________

9. Register fire extinguisher pressure: ______________________________

10. Verify that fire alarm glass is unbroken:

    □ OK

    □ NOK

    □ NA

11. Verify if all lights are operational:

    □ OK

    □ NOK - Comment if NOK: ______________________________

    □ NA

12. Verify that insect killer trap box is not full:

    □ OK

    □ NOK

    □ NA

Appendix E - Students@DETI Content

Alongside the poster, demonstration videos of the prototype developed up to the date of the event (07-06-2023) were presented[3].

An A4 version of the poster is printed on the next page.

[3] https://www.youtube.com/playlist?list=PLpEPS1zh4byel8i_GTORuu65dJnHvPhYn

Remote Assistance through Virtual & Augmented Reality for Maintenance Scenarios

Ivo Félix

Supervisor: Paulo Dias PhD

Co-Supervisor: Bernardo Marques PhD

Master's Dissertation in Informatics Engineering, 2nd year, MEI.

2023

Short Abstract

Today's globalized economy often produces situations for which an expert is required on-site but co-location is either not possible or convenient due to time constraints or to reduce costs. In this work, a remote assistance system for physical tasks making use of virtual and augmented reality was developed and evaluated in a building maintenance use case.

System Overview

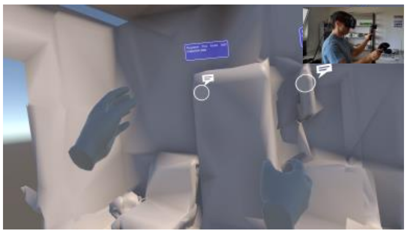
The system comprises two applications running on different devices: an AR application for the local technician on-site running on a Microsoft HoloLens 2 and a VR application for the remote expert running on either an HTC Vive or a Meta Quest 2. The remote expert is immersed in a 3D reconstruction of the local technician's environment, created using the depth sensors of the local technician's AR device, in which he produces step-by-step instructions using virtual objects, namely arrows, circles, comments, and real object replicas.

Spatial Referencing Strategy

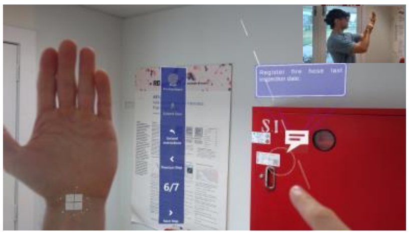
A QR code placed on the local technician's side is recognized by the device and the transformation that aligns the generated 3D reconstruction of the environment with the physical environment in relation to the QR code is computed and used to position the virtual objects created on the remote expert's side.

System Workflow

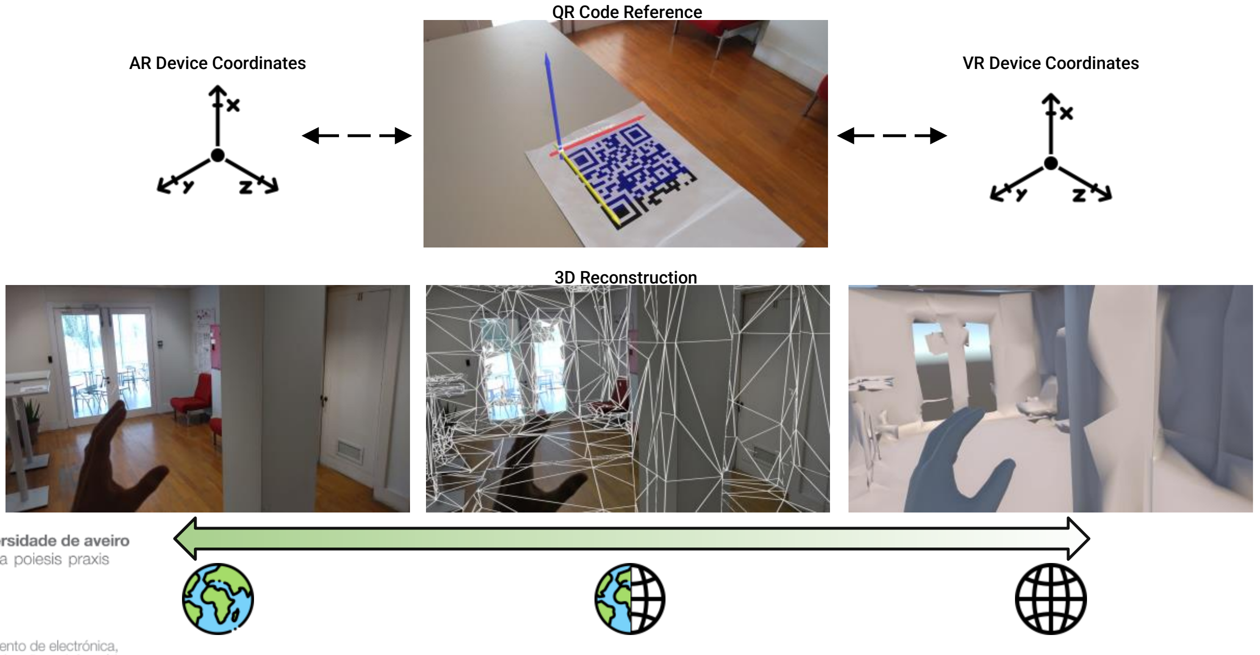
1. Local Technician Side – Start environment scan in the application and submit it.
2. Remote Expert Side – Create instructions in the loaded VR environment and submit them.



3. Local Technician Side – Load AR instructions and follow the steps.



Connecting the Real and Virtual World



universidade de aveiro
theoria poiesis praxis

deti departamento de electrónica, telecomunicações e informática