MASTER IN COMPUTER SCIENCE

Mixed Reality gestures for printer maintenance

MACHADO DOS SANTOS, Pedro

Award date:
2022

Awarding institution:
University of Namur

UNIVERSITÉ DE NAMUR

FACULTÉ D'INFORMATIQUE

Mixed Reality gestures for printer maintenance

Pedro Machado dos Santos

RUE GRANDGAGNAGE, 21 ● B-5000 NAMUR(BELGIUM)

Université de Namur
Faculté d'informatique
Année académique 2021–2022

Mixed Reality gestures for printer maintenance

Pedro Machado dos Santos

Promoteur : ______________________ (Signature pour approbation du dépôt - REE art. 40)
Bruno Dumas

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Acknowledgements

I would like to express my gratitude to my promoter, Prof. Bruno Dumas, for being available for any discussion, for your quick answers and feedback.

I also would like to thank all the participants who take part in the gesture elicitation study, for being available and taking the time to do it.

Abstract

We live in a world where immersive technology is ubiquitous in today's society. The most common types of immersive technology are Augmented Reality (AR), Virtual reality (VR) and Mixed Reality (MR). A great example of AR is Snapchat or Pokemon Go. Thanks to the availability of devices that make easier the access to this kind of technology, several studies were made in different fields, such as maintenance. This work aims to explore Leap Motion Controller hand detection capabilities to build a prototype for printer maintenance. We performed a user-elicitation study with 12 participants to find out their preferred gestures for a set of referents. Based on the resulting gestures, we present a complete user-defined gesture set and quantitative agreement rates. We implemented a mixed reality setup with a Leap Motion Controller and based user-defined gesture set, and test it with a series of printer maintenance tasks.

Keywords — mixed reality, immersive technology, mid-air interactions, hand gestures, maintenance, training

Contents

1 Introduction

We live in a world where immersive technology is ubiquitous in today's society. For a long time, we have been seeing it through science-fiction movies like Star Wars, Iron Man ou Minority Report, but nowadays it is everywhere.

The research and development of new technologies permitted the design of applications to create immersive experiences in various fields such as games, assembly, maintenance, training, remote collaboration, and others [Kaplan et al., 2021, Juraschek et al., 2018, Kim et al., 2019, Chalhoub and Ayer, 2018].

The most common types of immersive technology are Augmented Reality (AR), Virtual reality (VR) and Mixed Reality (MR). The most popular uses for AR/VR applications, is for gaming and social media. A great example of AR is Snapchat or Pokemon Go. For VR, most immediately recognizable component is the head-mounted display (HMD), like the Oculus Rift and PlayStation VR (PSVR).

Mixed Reality (MR) systems are not as widely available as VR and AR devices or software, most of the current use cases of mixed reality technologies are tailored to a specific task or objective and have different applications within different fields.
MR applications are in the top 10 ranked ICT technologies in 2020 [noa,] and research continues to find new applications of MR within different sectors. [Stretton et al., 2018] performed a review regarding the specialized use of MR in health care. [Al Janabi et al., 2020] performed a study about effectiveness of the Microsoft Hololens MR headset in minimally invasive surgery. [Kim et al., 2019] performed an investigation into support MR in learning programming. [Feick et al., 2018] designed a low-cost system supporting object-focused remote collaboration.

Industrial revolutions caused changes in technological, socioeconomic and cultural features, like maintenance management. The approach towards maintenance has changed from reactive towards predictive, tries to predict breakdowns and minimize the risks and costs associated with it. However, breakdowns still happen, and they need to be resolved (what to do when they occur).
There are many papers about performance impacts, and potential objective and subjective benefits of using MR in maintenance. [Espíndola et al., 2013] presented an approach to facilitate interaction with maintenance systems through intuitive interfaces, this is a competitive advantage in terms of time and costs for industry. [Utzig et al., 2019] presented a concept study to facilitate maintenance of an operating aircraft based on its lifelong collected data. [Aitken and Ross, 2019] conducted a study that compares the performance of automotive technician apprentices, who were conducting a routine vehicle maintenance task using the current training method, and MR training method. [Fonnet et al., 2017] presented HeritageCARE project that focus on prevention maintenance, and aim to improve building inspector work during their inspection routine.
MR can be used for remote assistance too. If a technician is having some trouble during the maintenance, he can call his supervisor to help him since he is viewing the same thing as the technician. Or when a company does not offer on-site support in a country.

Whether a printer is out of ink, missing paper or the paper is jammed, printer troubleshooting can be a pain. Now, with mixed reality, printer repair and support can be easier. The aim of this work is to propose a system of mixed reality maintenance on printers, using a Leap Motion Controller and a computer. The system uses mixed reality to assist the technician/user doing the maintenance tasks. The tasks are decomposed in several instructions that are displayed individually, using virtual objects. The proposed solution can be adapted to a more comfortable and ergonomic hardware like a HMD (i.e. the Hololens from Microsoft). We will be using hand gestures commands. Even some devices accept other types of commands, like by voice or eye tracking.

Gestures are often system-defined, not user-defined or field-defined or scenario-defined. Gestures offered by the system are not adapted to the field. The gestures needed in a surgical operation and in a printer maintenance are different. The first step is to design a set of hand gestures for our defined scenario. A user-elicitation study with 12 participants was performed to determine their preferred gestures for a set of tasks. That was accomplished by employing [Piumsomboon et al., 2013] elicitation methodology.
Using the agreement found among hand gestures, a gesture set was created and used to design user-centered gestures in our MR maintenance program. In this study, we focus explicitly on hand gestures for unimodal input.

The remainder of this paper is structured as follows: Section 2 presents some theoretical background. Section 3 presents the hand gesture elicitation study. Section 4 presents our printer maintenance design and Section 5 summarizes our research and proposes a new direction of research.

2 Background

2.1 Definition of AR, MR and VR

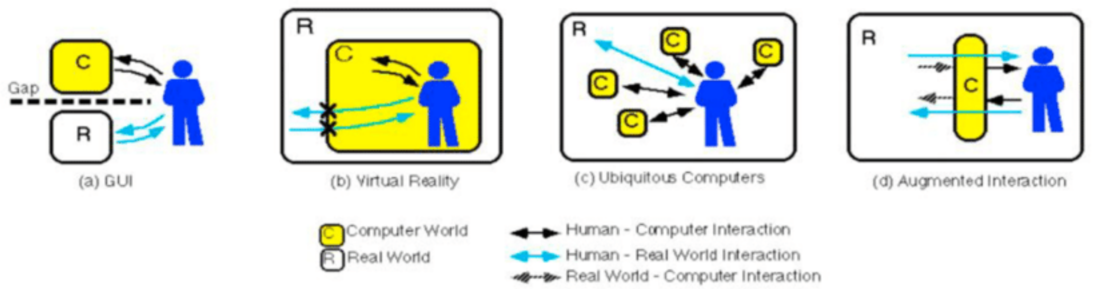
VR, AR, and MR are all types of immersive technology. Experts were more in agreement about what constitutes VR, it is well established and there is no fogginess about it. VR is an immersive and interactive simulated environment (fully synthetic or virtual) that is experienced in the first person. The user is isolated from the real world and nowdays, usually through immersive head-mounted displays (HMD) and hand-held controllers provide hand and body tracking.

Concerning the definition of MR and the difference between AR and MR, there is some fogginess. AR is commonly defined as a technology that allows the superimposition of virtual data upon the real world. Although there are different interpretations and taxonomies, like [Azuma, 1997, Milgram and Kishino, 1994, Rekimoto and Nagao, 1995].

According to [Azuma, 1997], augmented reality can be defined by the three distinct following criteria:

- Combination virtual and real environment,
- Real-time interaction between both environments,
- Registered in 3D space.

[Rekimoto and Nagao, 1995] proposed AR as a new type of human-machine interface, like GUIs (i.e. Graphical User Interface). In Figure 1, [Rekimoto and Nagao, 1995] represents augmented reality as "*Augmented Interaction*" (Figure 1d). It removes the gap between computers and reality found in a conventional GUI (Figure 1a).



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

Figure 1: Diagram of human-machine interactions, [Rekimoto and Nagao, 1995]

[Milgram and Kishino, 1994] proposed that augmented reality is in a continuum, with two extremes, a fully real environment and a fully virtual environment (Figure 2). Everything in between is described as MR. MR environments are defined as of real and virtual elements, between the two ends of the Virtuality Continuum, merged in a display.
The aim of MR is to merge real and virtual environments, where real and virtual objects coexist and interact in real-time with users. According to this definition, VR is not part of MR, and AR is only a subset of MR.

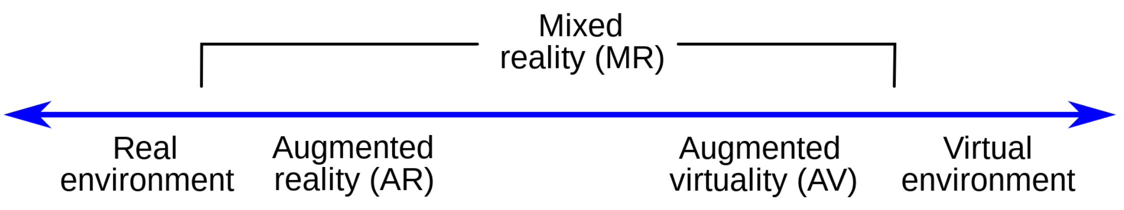


Figure 2: Virtuality Continuum from [Milgram and Kishino, 1994]

Since MR paradigm is the fuzzy one, [Parveau and Adda, 2018] proposed *3iVClass*, a new classification method for VR, AR and MR. *3iVClass* is based on [Azuma, 1997, Milgram and Kishino, 1994, Rekimoto and Nagao, 1995] taxonomies and is defined by the three following criteria:

Immersion: immersive experience description of the user experience.

Interaction: type of interaction provided by the technology.

Information: types of information and data managed.

Definition of MR according the *3iVClass* classification method:

Immersion: User environment must be processed and interpreted in real time (spatial mapping or spatial understanding). The user is at the center of his experience.

Interaction: User interactions (like gesture, voice, gaze and without any controller) must be processed in real-time.

Information: Virtual objects must be registered in space and time. Virtual objects need to be displayed three dimensions, according to the user position, the environment, or any other objects.

Figure 3 presents the three essential components of MR technology. The association between the man/machine interactions, the computer perception of the environment and the real environment reality form together the MR.

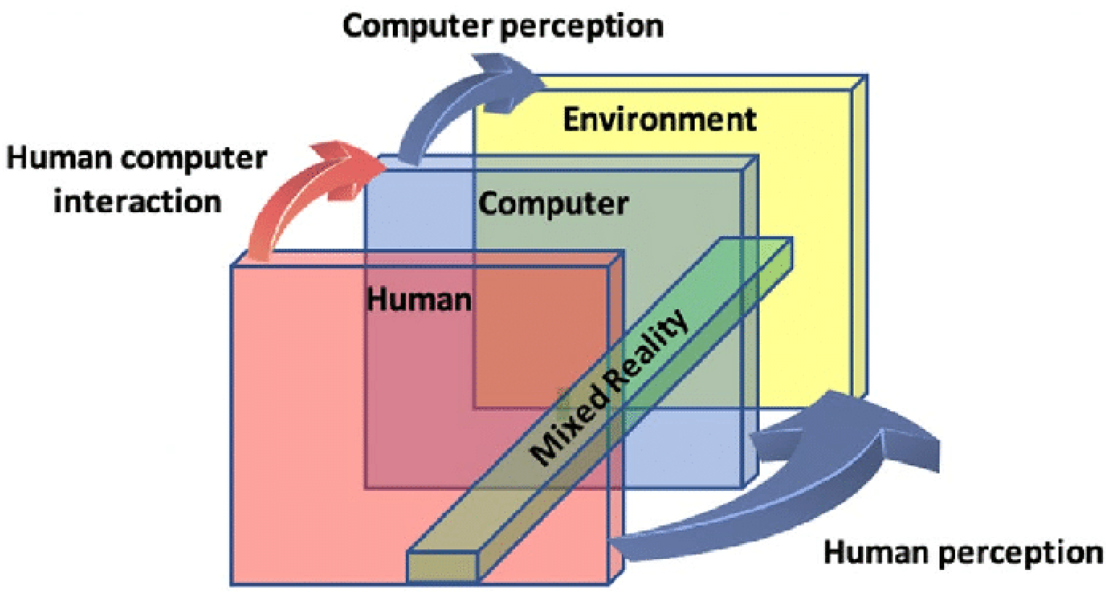


Figure 3: Mixed reality principal components, [Parveau and Adda, 2020]

The *3iVClass* method defines MR as follows: *"Mixed reality is a paradigm that combines technologies that, by mapping the user's space, display 3D-embedded virtual content registered in space and time. Virtual objects can be positioned relative to the real environment, the user, or any other virtual or physical object. In addition, the mixed reality experience must be user-centered and offer natural and immediate interactions"*.

[Speicher et al., 2019] presented a shared understanding of the term MR (concepts and technologies), through interviews (AR/VR experts from academia and industry) and a literature survey.
They were able to identify six different notions of MR, from the literature survey and expert interviews:

1. Continuum : the Reality–Virtuality Continuum from [Milgram and Kishino, 1994]
2. Synonym : MR as a synonym for AR.
3. Collaboration : MR as a type of collaboration.
4. Combination : MR as a combination of AR and VR.
5. Alignment : MR as an alignment of environments.
6. Strong AR : MR has a "stronger" version of AR.

The study presented a conceptual framework for classifying MR experiences along seven dimensions, classified by the six notions of MR derived from the literature survey and expert interviews, as we can see in Figure 4.

Dimension	**# Environments**		**# Users**		**Level of Immersion**			**Level of Virtuality**			**Interaction**		**Input**	**Output**
value	*one*	*many*	*one*	*many*	*not*	*partly*	*fully*	*not*	*partly*	*fully*	*implicit*	*explicit*	*any*	*any*
1—Continuum	✔		✔		✔	✔	✔		✔	✔	✔		✔	✔
2—Synonym	✔		✔		✔	✔			✔		✔		✔	✔
3—Collaboration	✔	✔		✔		✔	✔		✔	✔	✔	✔	✔	✔
4—Combination	✔		✔			✔	✔		✔	✔	✔		✔	✔
5—Alignment		✔	✔		✔	✔	✔	✔	✔	✔	✔		✔	✔
6—Strong AR	✔		✔			✔			✔		✔	✔	✔	✔

Figure 4: Conceptual framework for classifying MR experiences along seven dimensions from [Speicher et al., 2019]

The *continuum* notion, according to [Milgram and Kishino, 1994], is the most used (35.3%).

2.2 Interaction technologies

MR systems have several types of input methods. Different input methods have been used for different types of MR applications, depending on the application user interaction tasks required. Interaction methods in MR are based on methods implemented in AR environment, since researchers and experts are considering AR has a subset to MR technology, according [Milgram and Kishino, 1994] definition.

[Billinghurst et al., 2015] presented a number of interface types developed for AR, such as :

1. Information Browsers: Interfaces displaying AR information on the real world, i.e. AR browser Wikitude.

2. 3D User Interfaces: Manipulate content in space through controllers, i.e. 6DOF joysticks.
3. Tangible User Interfaces: Using real objects to interact with AR virtual content.
4. Natural User Interfaces: Using natural body input like hand gestures to interact with virtual content, i.e. Microsoft Kinect.
5. Multimodal Interfaces: Combined input, being speech and gesture one of the most widely researched combination, i.e. Microsoft HoloLens.

The interaction methods in AR have been thoroughly studied, with the arrival of new hardware technologies, the study of multimodal interaction in MR begun.

[Link et al., 2016] presented a mixed reality tabletop role-playing games with real-time interaction. Game characters are commanded by high-level commands via a multimodal interface (speech and gesture). [Moniri et al., 2016] proposed a system that enables collaboration between a robot, a human, and a remote human. This hybrid team can interact and cooperate with and within the system using eye, head, hand and gesture tracking. [Zimmer et al., 2017] presented design and implementation issues of mixed reality prototypes using the Microsoft Hololens. The prototypes allowed several input methods such as voice input, gesture recognition and spatial input, which are features of the Microsoft Hololens HMD. [Jacucci et al., 2018] proposed an application to help tourists by providing digital information. User can use gesture, motion and touch interaction techniques to interact with the digital information.

2.3 Frameworks

Several frameworks can be found in literature. Many frameworks, for multimodal interface (MMI), have been proposed to guide the designers and developers to design multimodal applications (in augmented reality environment as well).

[Abidin et al., 2017] presented a framework to illustrate the adaptive multimodal interface for location-based augmented reality application. It was developed by combining components from other frameworks (Figure 5) in augmented reality, multimodal interface and adaptive interface field.

	[Kong et al., 2011]	[Duarte and Carriço, 2006]	[Möller et al., 2014]	[Minkyung Lee,]	[Solórzano,]	[Dumas et al., 2009]	[Lee and Billinghurst, 2013]
Multimodal Interfaces	X	x	x	x	x	x	
Augmented Reality				x			x
Adaptive Interfaces	X	x	x		x		
Input Modalities	**Hand gesture, voice, Multi-touch display etc.**	**Speech, Mouse & Keyboard Input**	**Gesture, Touch, Speech**	**Speech, 3D hand gesture**	**NFC Tag, Speech, 2D gesture**	**Speech, Gesture**	**Not Focus**
Output Modalities	Audio, Video, Graphics etc.	Visual Display and sound	Vibration, Sound, Visual	Visual	Not Focus	Not Focus	Not Focus
Task	Social Networking Application	Digital Talking Book (DTB)	3 Application Demo	MMI in Table top environment	Multimodal Adaptive Agenda (MAA)	IM2	CourierAR, CityViewAR, GeoBoids
Application Domain	Service	Entertainment	Education	Education	Service	Service	Service, Education and Entertainment

Figure 5: Comparison of different types of frameworks, [Abidin et al., 2017]

[Fernández del Amo et al., 2018] focus in maintenance, and presented a framework that analyses requirements for an integration of AR in industrial maintenance systems. Regarding AR authoring (development tool allowing to create, edit and update AR contents [Zhu et al., 2013]) for maintenance, the four main challenges found are:

Authoring tools users and ease-of-use: the types of authoring and the skills required to use authoring tools. The two main trends in literature are the use of easy GUI's to generate low-cost applications, the focus on the user type and when they would be able to create content (on-site, online and offline authoring). However, all papers mention animations as a type of data to be rendered to maintenance operators.

Visualisation challenges: appear when using animations to explain complex tasks. [Gimeno et al., 2013] *Sugar* tool tries to solve occlusion or photorealism, allocating virtual content in the most effective way, considering different AR techniques.

Information display modes: define the displayed information and format to increase maintainers performance. Re-using information helps avoid data duplication and reduce AR implementation costs.

Interaction with the virtual data displayed: types of interactions allowed between users and applications, such as multimodal interaction, gesture interaction or 2D and 3D visualizations. When using applications, the type of interaction modify how the data should be displayed, but also what data can be captured from maintainers.

Figure 6 presents the maintenance challenges, consequences and its triggers, recognized by [Fernández del Amo et al., 2018]. Looking to the triggers, current support given to technicians is not always enough for the complex tasks. The author concluded that continuous improvement tools are not used nor exist if the errors continuously increase and the performance decrease.

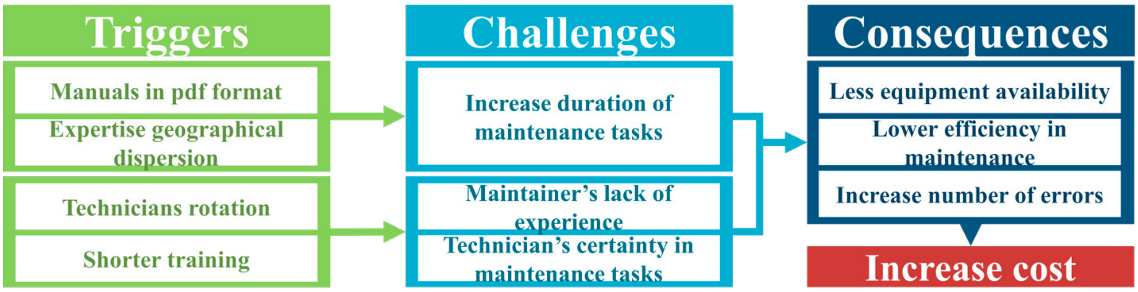


Figure 6: Maintenance challenges, consequences and triggers operations, [Fernández del Amo et al., 2018]

The framework is created by structuring a classification, as we can see in Figure 7, with the information required and its main characteristics for each maintenance process (diagnosis, repair and analysis).

	Authoring		Context-Awareness		Interaction
Repair	Information	Instructions	Context	User-centred	3D and 2D visualisation
	Format	Animations and text	Data	Maintenance skills	
Diagnose	Information	Equipment's health condition	Context	Equipment-centred	3D and 2D visualisation
	Format	Coloured 3D models and text	Data	Sensors' information	
Analyse	Information	Performance time and errors	Context	User and equipment centred	Gesture interaction and 3D visualisation
	Format	Text	Data	Sensors' information	

Figure 7: Maintenance information needs - formats and content classification, [Fernández del Amo et al., 2018]

To help maintainers diagnosing faults, the framework uses 3D colored models based on the sensor data of the equipment tracked. That data can be used to verify if the technicians performed the diagnosis correctly. Gestures can be tracked to assess technician performance while repairing. To improve the processes, it is proposed to use the performance data captured during diagnosis and repair processes. [Fernández del Amo et al., 2018] framework allows developing maintenance applications with different levels of immersion and interaction between virtual and real world. Increasing interaction contributes to acquire more data and more knowledge that can be transferred between users and systems.

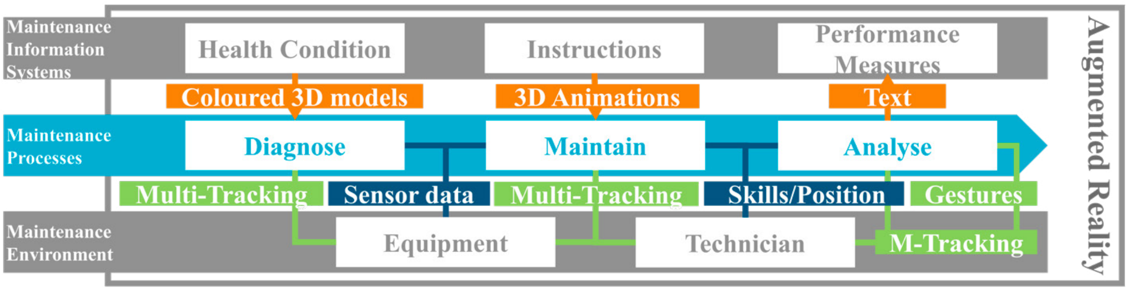


Figure 8: Information framework for AR integration in maintenance systems, [Fernández del Amo et al., 2018]

We can find many frameworks for designing mixed reality systems with different input methods and in different domains.

[Rokhsaritalemi et al., 2020] presented a generic framework, containing all MR components needs, composed of five layers: concepts, systems architecture, middle-ware, applications, and user interface (UI).

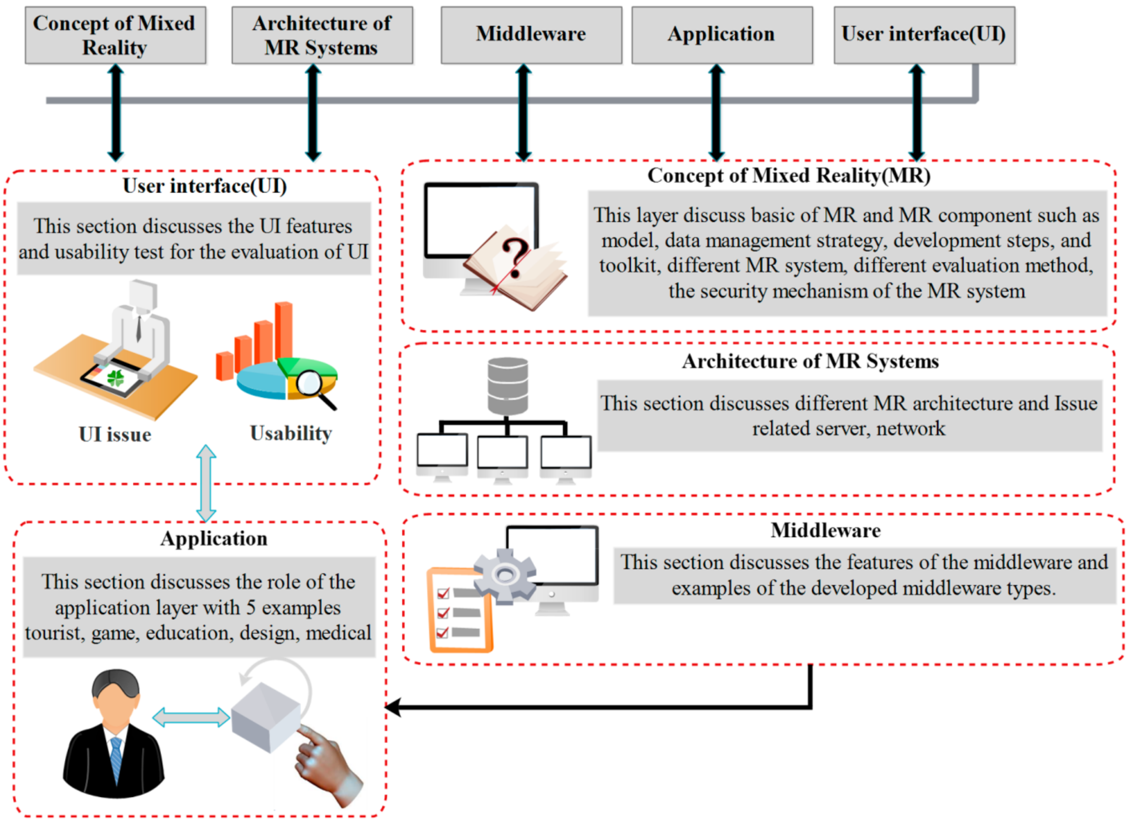


Figure 9: Research layers, [Rokhsaritalemi et al., 2020]

The first layer is composed of MR components such as algorithms, data management, development guidelines, frameworks, evaluations, security and privacy. The architecture layer expresses architecture, server, and network issues. Middle-ware layer introduces the features for MR support, and provides examples of the developed middle-ware types. MR application gives services to users, by providing connectivity to sensing and MR devices (i.e. HMD) and enables user interaction (can be multimodal) by using an UI. The UI layer introduces UI features, and conventional techniques used to evaluate user satisfaction, usability testing for MR.

2.4 Hardware

Mixed reality are emerging interactive and display technologies, and the devices follow. [Park et al., 2020] consider MR as synonymous with AR and grouped the devices into multiple, distinct subtypes (Figure 10). Head-mounted displays are categorized into optical see-through (OST) and video see-through (VST).

On OST displays we have a direct view of the external environment, they use special transparent lenses. OST do not cut users from the physical world, unlike VST. On VST displays we have an indirect view of the external environment, a video feed is used to view the external environment. Monocular are devices with a single display and binocular devices with dual displays, project the image directly onto two see-through lenses/screens.

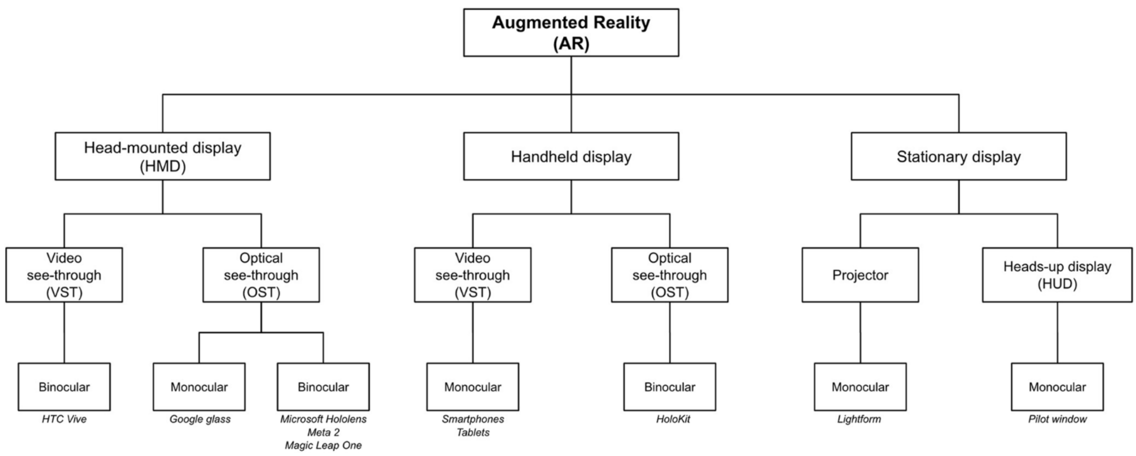


Figure 10: Types of AR devices (examples in italics), [Park et al., 2020]

2.4.1 Head-Mounted Display

Nowadays, optical see-through HMDs are at the cutting-edge of the MR research, and several headsets have been developed following the success of the Microsoft HoloLens 1 in 2016, such as the HoloLens 2 (Figure 11a), the MagicLeap One (Figure 11c), the Lumus DK-Vision or the Lenovo ThinkReality A6.

The successful use of these devices in practical applications still limited by the complexity and unreliability of the calibration procedures [Qian et al., 2017]. Calibration is needed to ensure an accurate spatial alignment between the real-world and the virtual elements rendered onto the see-through display. Several studies have reported the accuracy of the Microsoft HoloLens to be near or within one centimeter [Incekara et al., 2018, Gibby et al., 2019, Rae et al., 2018].

(a) Microsoft HoloLens 2, [Microsoft,]

(b) Microsoft HoloLens 2, [Microsoft,]

(c) Magic Leap 1, [Leap,]

Figure 11: Mixed reality optical see-through HMDs examples

As for OST, there are several video see-through HMDs such as the Varjo XR-3 (Figure 12a), HTC Vive Pro (Figure 12b) or the Lynx R1 (Figure 12c).

(a) Varjo XR-3, [Varjo,]

(b) HTC Vive Pro, [HTC,]

(c) Lynx R1, [Lynx,]

Figure 12: Mixed reality video see-through HMDs examples

2.4.2 Hand tracking

Experimental setups improve the understanding of the underlying subject, but require special hardware. OSTs are a type of hardware, but can be expensive.

The Leap Motion Controller (LMC) is a commercial low-cost optical hand tracking that captures the motion of a human hand in three dimensions by recording various parameters. Leap Motion Controller has multiple usages such as productivity applications with computers, integrated hardware solutions or displays, or attached to headsets for prototyping, research, and development.

Figure 13: Ultraleap Leap Motion Controller, [UltraLeap, b]

From a hardware perspective, the device is composed of two cameras and some infrared LEDs, which track infrared light at a wavelength of 850 nanometers (outside the visible light spectrum). The Leap Motion Controller has an interaction zone from 10cm to 80cm (60 recommended) above the device, and an action area about 140° by 120° wide (Figure 15).

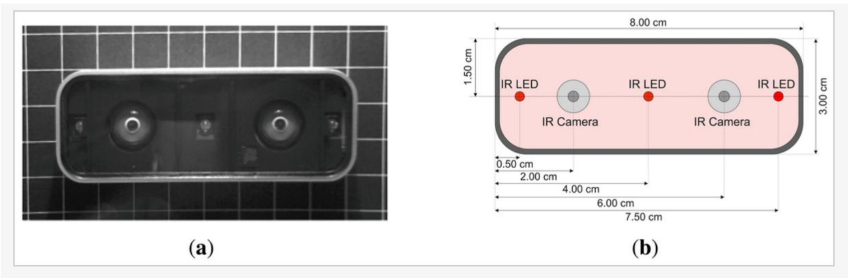
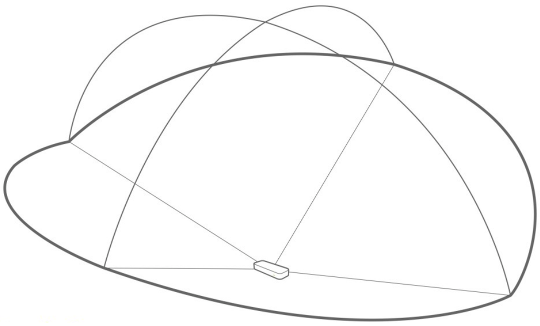
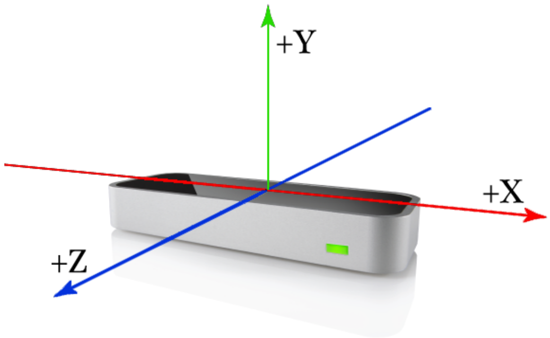


Figure 14: Leap Motion Controller internal structure, [Shao and Stanford, 2016]

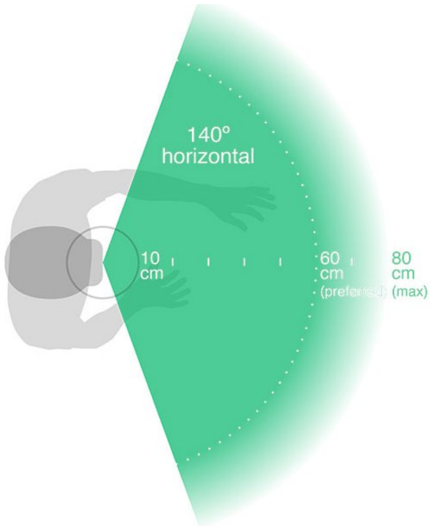
The tracking precision has been proved to be accurate, the results showed position deviations to be lower than a millimeter. Although it has been noted by [Weichert et al., 2013, Guna et al., 2014, Vysocky et al., 2020], that when making measurements in the dynamic environment, accuracy drops with higher distances, when the measurements were made outside the interaction zone.



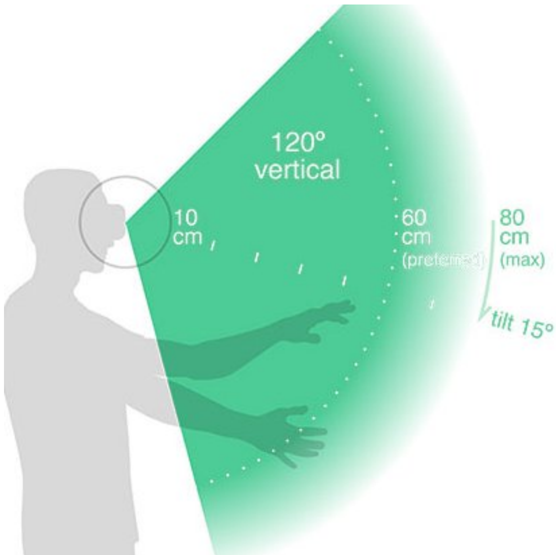
(a) Interaction area, [LeapMotion, 2015]



(b) Interaction area, [UltraLeap, b]



(c) Horizontal interaction area, [UltraLeap, b]



(d) Vertical interaction area, [UltraLeap, b]

Figure 15: Leap Motion interaction zones.

The Leap Motion tracks hands, fingers, and tools in its field of view. The LMC SDK can discern 27 distinct hand elements and provides access to all fingers and its bones data (Figure 16), which makes it simple to control a virtual representation of the user's hand.

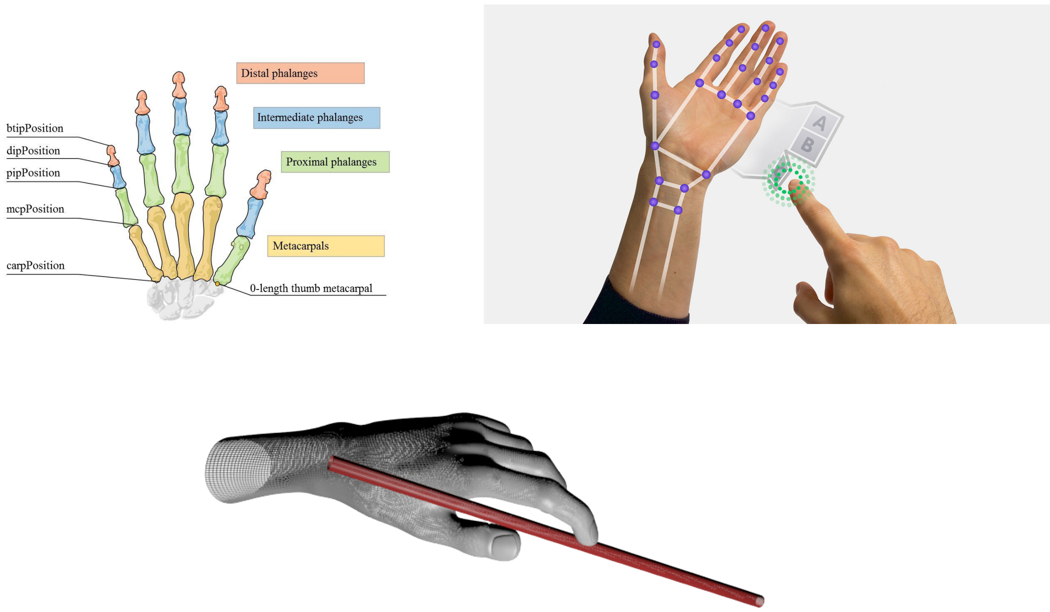


Figure 16: Skeletal tracking mode, [UltraLeap, a].

The last SDK, know as Gemini, provides a compatibility with Unity, Unreal, OpenXR and MRTK. The LMC is used by several HMDs.

2.5 Software

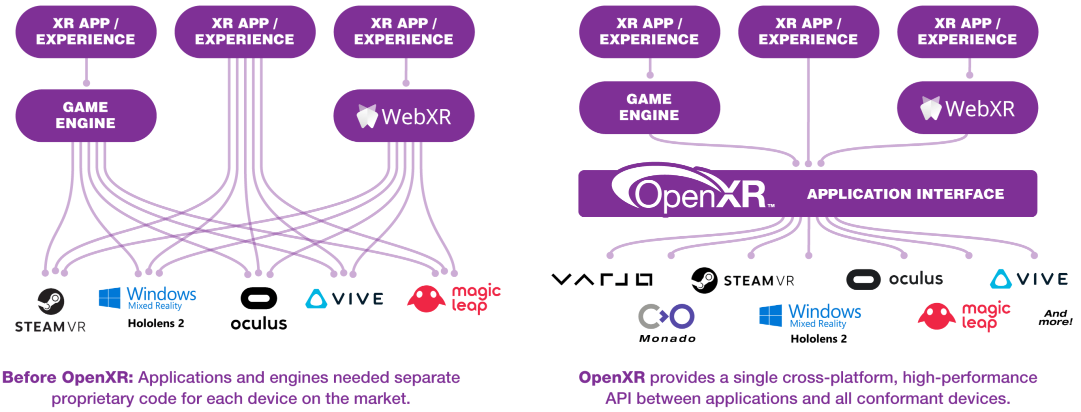
To build experiences that seamlessly blend the digital and physical worlds, programming languages and simulation tool-kits are needed for developing the MR platforms.

[Rokhsaritalemi et al., 2020] summarized software development, tool-kits, and the programming interface (Figure 17) available in 2017. Some appeared after like MaxST and ARcore, but are not limited to.

Software Toolkit	Detection or Tracking	Platform	Development Language	Licensing	Current Version
AR Toolkit	2D objects	Android, iOS, Linux, macOS, Unity3D	C, C++, C#, Java	Free open source	ARToolKit6 (2017)
Layar	2D objects	Android, iOS, Blackberry	HTTP(RESTful), JSON	Paid	Layar v7.0
Vuforia	3D objects, marker based visual SLAM	Android, iOS, Windows (selected devices), Unity 3D	C++, C#, Objective-C, Java	Free (with paid versions)	Vuforia 6.5
Zapbox	3D objects, marker based visual SLAM	Android, iOS	JavaScript	Paid (with 30-day free trial)	Zapworks Studio (2017)
EasyAR	3D objects, visual SLAM	Android, iOS, macOS, Windows, Unity3D	C, C++, Objective-C, Java	Free (with paid version)	EasyAR SDK 2.0
kudan	3D object, visual SLAM	Android, iOS, Unity 3D	Objective-C, Java	Free (with paid versions)	kudan v1.0
Wikitude	3D objects, visual SLAM	Android, iOS, Smart Glasses (EpsonMoverio, ODG, Vuzix), Unity3D	Objective-C, Java, JavaScript	Paid	wikitude SDK 7.0
ARkit	3D objects, visual and depth SLAM	iOS	Objective-C	Free	ARKit (2017)
OSVR	2D objects, orientation-based SLAM	Vuzix, OSVR, HTC Vive, Oculus, SteamVR	C, C++, C#, JSON for plug-ins)	Free (opensource)	OSVR Hacker Development Kit (HDK) 2.0
Meta 2 SDKs	3D objects, visual and depth SLAM	Meta 2 glasses	C#, C++, JavaScript	Free	Meta 2 SDKs
Windows Mixed Reality	3D objects; visual, depth, orientation-based SLAM	Windows MR, HoloLens, Unity	C#, C++, JavaScript	Free	Windows MR API

Figure 17: MR software, toolkits, and application programming interface, [Rokhsaritalemi et al., 2020]

Some software is linked to a platform or a device, with makes difficult creating cross-platform mixed reality experiences. Khronos provides OpenXR, which is an open royalty-free API standard that addresses the most common and challenging aspects of MR development. It provides engines with native access to a range of devices across the mixed reality spectrum. Code is portable across a wide range of hardware platforms. Figure 18 shows the before/after OpenXR. Before several branches of the application or the game engine were needed to be compatible with different hardware.



(a) Before OpenXR (b) After OpenXR

Figure 18: OpenXR: Solving XR fragmentation, [Khronos, 2016]

2.5.1 WebXR

Mixed Reality features are available on the Web through WebXR. It is possible to see virtual reality (VR) and augmented reality (AR) content in a compatible WebXR-enabled browser. WebXR Device API allows the development and hosting of VR and AR experiences on the web, providing access to input (information from headset and controllers) and output (hardware display) capabilities.

2.5.2 Game engines

A game engine is a software framework where the core functionality includes a rendering engine for 2D or 3D graphics, a physics engine or collision detection, sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and video support for cinematics.

Game engines were primarily designed for the development of video games, and due to the gaming market increase, several frameworks were developed. A not exhaustive list can be found on [Wikipedia, 2022].

There are dozens of game engines, but not all of them are suited to MR. Not all of them are compatible with MR hardware (i.e. the Leap Motion only provide plugins for Unity and Unreal). The two most supported game engines in mixed reality are Unity and Unreal.

Unity is a cross-platform game engine written in C++ and the development scripting is done in C#. Unreal is an open source creation engine with full support for mixed reality in both C++ and Blueprints. Both, Unity and Unreal, support a variety of desktop, mobile, console and virtual reality platforms.

2.6 Maintenance

The ultimate goal of each business owner is to maximize productivity and minimize costs. Therefore, a strategy to evaluate the average cost of failure removal compared to average maintenance costs is needed.

Maintenance has gone through the stages of revolution, and can be broken down into four categories, reactive, preventive, productive and predictive.

Reactive maintenance, is an unscheduled corrective maintenance, performed to deal with a malfunction or incident, in order to restore the production to the desired state (i.e. replacing broken machine parts, debugging after a software error). Reactive maintenance itself can be relatively low cost, but the cost of stopping an operation can be expensive.

Preventive maintenance is the regular and routine maintenance of equipment and assets, it focuses on the prevention of failures and incidents. The replacement or repair of the equipment during regular inspections before it fails, prevents unplanned downtime and related costs from unexpected equipment failure.

Productive maintenance is referred to as a strategy to involve everyone working in a facility in maintenance, rather than just the maintenance team. It makes production and maintenance teams work together and be equal partners. The philosophy of productive maintenance is not only to prevent failures, it has some direct and indirect benefits such as reducing errors, unplanned downtime, manufacturing costs short-term downtime, shorten the time of change of product, *etc.*

Predictive maintenance uses data science and predictive analytics to estimate when a piece of equipment might fail and schedule corrective maintenance operation before the point of failure. Maintenance is scheduled at the most convenient and most cost-efficient moment, optimizing the equipment lifespan to its fullest, but before the failure. It means, understanding how everything works and is related, including the environment. Often this knowledge can be gathered from the equipment manufacturer documentation and the maintenance team experience who have worked on similar systems for a long time. To evaluate equipment condition, predictive maintenance measures and gathers operations and equipment real-time data, using nondestructive testing technologies such as infrared, acoustic, corona detection, vibration analysis, sound level measurements, oil analysis, thermal imaging, and other specific online tests.

2.7 Related work

The topic of mixed reality in printer maintenance have already some implementations such as [Novikov, 2019]. The author shows the changes in maintenance management strategies and with his proof of concept, demonstrates with a modern technology an effective solution for problems that have already arisen. The solution proposed, assumes the use of HoloLens from Microsoft. However, the system uses more of a WIMP interaction (click on buttons) rather specific mid-air commands for some tasks.

[Krings et al., 2020] proposed a modular development framework *AARCon* which eases the development of context-aware applications for mobile AR. In mobile AR, context changes often happen, due to user movements and manipulating their devices in different environments and situations. Context-awareness makes an AR application adjust to the user, their situation and needs, making the application more ergonomic and easy to work with. Their case study, presents the creation of a context-aware AR printer maintenance application to show the potential of AARCon.

Even the previous works are about AR/MR printer maintenance, none uses advanced gestures as commands. In this study, we propose some mid-air gestures to accomplish some tasks instead of the well-known WIMP interaction (click on buttons). Since gestures are often system-defined, not user-defined, field-defined or scenario-defined, we realized a gesture elicitation to design a set of hand gestures based on user and field.

2.8 Gesture elicitation

Gesture interaction has some problems like imposition, the lack of acceptability and the lack of consensus, basically gestures are often system-defined, not user-defined.

Gesture elicitation is a technique where the desired effect of an action (called a referent) is shown to the user, and he proposes a gesture (called a symbol) that would cause the effect to happen. The results from all users are merged to create a single gesture set (with or without synonyms), using metrics such as agreement, max-consensus, or consensus-distinct ratio.

Many studies were made about users preferences for finger, hand, wrist, arm, head, leg, foot, and whole-body gestures. Gesture elicitation studies (GES) have demonstrated their value as human–computer interaction (HCI), controlling all kinds of interactive devices, applications, and systems.

[Villarreal-Narvaez et al., 2020] deliver a systematic literature review about 216 GES by summarizing the characteristics and findings. As seen in Figure 19, heat-map rendering the number of GES for a specific body parts, hand gestures are the most elicited.

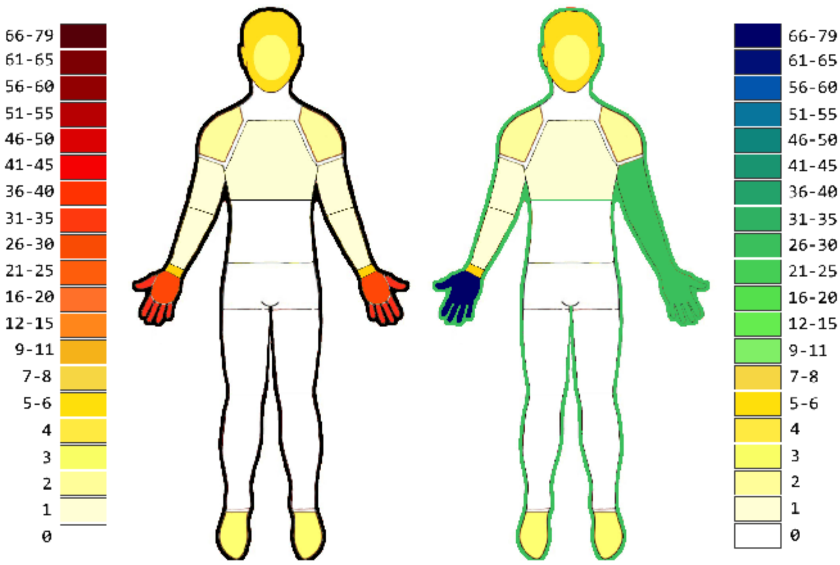


Figure 19: Heatmap representations of the number of GES. In the left, gestures involving one body part only, and in the right, gestures produced as the combination of at least two body parts, [Villarreal-Narvaez et al., 2020]

.

[Villarreal-Narvaez et al., 2020] have also listed the most influential GES (Figure 20), according Google Scholar citation's numbers. [Wobbrock et al., 2009] is the most cited, but it is based on the guessability study methodology of [Wobbrock et al., 2005].

	Authors	Study	Application	Year	Cita-tions
1.	Wobbrock *et al.* [123]	User-defined Gestures for Surface Computing	interactive tabletops	2009	1070
2.	Ruiz *et al.* [90]	User-defined Motion Gestures for Mobile Interaction	smartphones	2011	373
3.	Kane *et al.* [47]	Usable Gestures for Blind People: Understanding Preference and Performance	tablets, smartphones	2011	256
4.	Wobbrock *et al.* [122]	Maximizing the Guessability of Symbolic Input	hand-stroke gestures	2005	253
5.	Morris *et al.* [72]	Understanding Users' Preferences for Surface Gestures	interactive tabletops	2010	228
6.	Vatavu [111]	User-defined Gestures for Free-hand TV Control	smart TVs	2012	169
7.	Piumsomboon *et al.* [81]	User-defined Gestures for Augmented Reality	Augmented Reality	2013	165
8.	Kühnel *et al.* [51]	I'm home: Defining and evaluating a gesture set for smart-home control	smart homes	2011	156
9.	Lee *et al.* [53]	How Users Manipulate Deformable Displays As Input Devices	unconventional displays	2010	144
10.	Nacenta *et al.* [73]	Memorability of Pre-designed and User-defined Gesture Sets	interactive tabletops	2013	143
11.	Cauchard *et al.* [17]	Drone & Me: An Exploration into Natural Human-Drone Interaction	human-drone interaction	2015	134
12.	Frisch *et al.* [38]	Investigating Multi-touch and Pen Gestures for Diagram Editing on Interactive Surfaces	surface computing	2009	125

Figure 20: Top 12 GES, authors, and application domains (according Google Scholar citations), [Villarreal-Narvaez et al., 2020]

2.8.1 Legacy bias

[Morris et al., 2014] proposed three techniques (can be used alone or in combination) for reducing legacy bias in gesture elicitation studies. The paper refers to legacy bias as the gesture proposals by the users being biased by their prior experiences with interfaces and technologies, such as WIMP (windows, icons, menus, and pointing). Those three techniques are :

1. production: requires users to propose multiple gestures for each referent, forcing them to move beyond simple legacy inspired techniques.
2. priming: expose users to a stimulus, to influence behavior in subsequent task. Fill in the users about the capabilities of the new interaction technology.
3. partners: invite users to participate in group studies rather than individually. Brainstorming is common to produce ideas or solve problems.

2.8.2 Closed and open elicitation

A gesture elicitation can be *open* or *closed*. In an *open elicitation* the participants propose the symbol for the referent, the symbol set is unlimited, keeping the process as natural and creative as possible. Since, too many symbols are allowed, it can be difficult to have a consensus. In a *closed elicitation* participants select symbols from a predefined set, which constrains their choice. Limiting the choice of the participants tends to increase agreement.

[Cafaro et al., 2018] proposes to do an open elicitation before a closed elicitation. After doing an open elicitation and computing the initial agreement, participants enter into a closed elicitation to converge towards a better agreement (or other measure), limiting the consensus set size.

2.8.3 Reverse elicitation

Eliciting input from end-users to design system interactions is a popular design method. Such studies are usually conducted in a lab, with a number and diversity of participants limited, and their representative results. To address the limitations above, [Ali et al., 2019] created *Crowdlicit*, a system to conduct elicitation and identification studies online. The system aims to increase the scalability, accessibility, and efficiency of elicitation and identification studies. The paper presents the *end-user identification method*, which reverses the elicitation study methodology, a symbol is shown to the user and he is asked to suggest the referent.

2.8.4 Measures

From [Villarreal-Narvaez et al., 2020] review, agreement is the most used measure. Agreement is mostly computed by scores [Wobbrock et al., 2009] and rates [Vatavu and Wobbrock, 2015, Vatavu and Wobbrock, 2016], but alternatives exists, such as [Madapana et al., 2018, Tsandilas, 2018, Wu et al., 2019, Vatavu, 2019]. Agreement is, progressively being replaced by disagreement computation [Vatavu, 2019].

2.9 Gesture taxonomies

[Wobbrock et al., 2009] proposed a four-dimensional taxonomy of surface gestures based on user behavior: *form*, *nature*, *binding*, and *flow*. Each dimension contains multiple categories, as shown in Figure 21.

The *form* dimension is unimanual, if two-handed gesture, it is applied separately to each hand.

The *nature* dimension was divided into four categories: *symbolic*, *physical*, *metaphorical* and *abstract*. Symbolic gestures are visual representations, like thumbs-up and thumbs-down for accept and reject. Physical gestures have the same effect on a virtual object as on a physical object (grabbing a virtual block and moving it). Metaphorical gestures act on, with, or like something else. For example, pointing an index finger forward and spinning it clockwise to indicate play or increase speed. Any arbitrary gestures were considered abstract, have no symbolic, physical, or metaphorical connection to their referents (tasks).

The *binding* dimension has four categories: *object-centric*, *world-dependent*, *world-independent* and *mixed dependencies*. Object-centric gestures only require information about the object being manipulated, e.g. scale an object with two fingers together on top. World-dependent gestures are defined within respect to the physical work-space, such as dragging an object off-screen. World-independent gestures can occur anywhere, except on temporary objects that are not world features. Mixed dependencies occur for gestures performed across multiple spaces, world-independent in one respect, but world-dependent or object-centric in another. This occurs for 2-hand gestures.

The *flow* dimension has two categories: *discrete* and *continuous*. A gesture is discrete when it is completed (performed, delimited, recognized, and responded), like select a menu. A gesture is continuous when ongoing recognition is required, like re-size gestures.

Taxonomy of Surface Gestures		
Form	*static pose*	Hand pose is held in one location.
	dynamic pose	Hand pose changes in one location.
	static pose and path	Hand pose is held as hand moves.
	dynamic pose and path	Hand pose changes as hand moves.
	one-point touch	Static pose with one finger.
	one-point path	Static pose & path with one finger.
Nature	*symbolic*	Gesture visually depicts a symbol.
	physical	Gesture acts physically on objects.
	metaphorical	Gesture indicates a metaphor.
	abstract	Gesture-referent mapping is arbitrary.
Binding	*object-centric*	Location defined w.r.t. object features.
	world-dependent	Location defined w.r.t. world features.
	world-independent	Location can ignore world features.
	mixed dependencies	World-independent plus another.
Flow	*discrete*	Response occurs *after* the user acts.
	continuous	Response occurs *while* the user acts.

Figure 21: Taxonomy of surface gestures ("w.r.t." means "with respect to"), [Wobbrock et al., 2009]

[Piumsomboon et al., 2013] extended the [Wobbrock et al., 2009] taxonomy to AR, by adding two more dimensions, *symmetry* and *locale* (see Figure 22).

The *form* dimension was kept unimanual. Two categories were discarded as they were not relevant to AR gestures: *one-point touch* and *one-point path*.

The *symmetry* dimension was introduced to classify gestures depending on numbers of hands used, one-handed (unimanual) or two-handed (bimanual). The unimanual was split into dominant and nondominant and bimanual was subdivided into symmetric (both hands execute the same form, i.e. scaling) and asymmetric gestures (forms of the hands are different).

The *locale* dimension (where the gestures are performed), has two categories: *on-the-surface*, *in-the-air* and *mixed locales.*

Taxonomy of Gestures in AR		
Form	static pose	Hand pose is held in one location.
	dynamic pose	Hand pose changes in one location.
	static pose and path	Hand pose is held as hand relocates.
	dynamic pose and path	Hand pose changes as hand relocates.
Nature	Symbolic	Gesture visually depicts a symbol.
	physical	Gesture acts physically on objects.
	metaphorical	Gesture is metaphorical.
	abstract	Gesture mapping is arbitrary.
Binding	object-centric	Gesturing space is relative to the object.
	world-dependent	Gesturing space is relative to the physical world.
	world-independent	Gesture anywhere regardless of position in the world.
	mixed dependencies	Gesture involves multiple spaces.
Flow	Discrete	Response occurs after the gesture completion.
	continuous	Response occurs during the gesture.
Symmetry	dominant unimanual	Gesture performed by dominant hand.
	nondominant unimanual	Gesture performed by nondominant hand.
	symmetric bimanual	Gesture using both hands with the same form.
	asymmetric bimanual	Gesture using both hands with different form.
Locale	on-the-surface	Gesture involves a contact with real physical surface.
	in-the-air	Gesture occurs in the air with no physical contact.
	mixed locales	Gesture involves both locales.

Figure 22: Taxonomy of gestures in AR extended from taxonomy of surface gestures, [Piumsomboon et al., 2013]

3 Gesture elicitation study

3.1 Overview

To identify intuitive and user-friendly gestures for interacting with our mixed reality maintenance system, we conducted a user gesture elicitation study as a part of the design process. Those gestures will be used to command the system.

The elicitation process follows the style of the guessability method from earlier studies [Wobbrock et al., 2009, Piumsomboon et al., 2013, Morris et al., 2014, Vatavu and Wobbrock, 2015, Vatavu and Wobbrock, 2016], where users are asked to define the gestures (symbol) they think are the most intuitive for achieving the proposed tasks (referent). And based on all proposed gestures, identifying common gestures by investigating consensus among the participants. The consensus gestures set is used for designing our mixed reality maintenance system (Section 4).

3.2 Participants

We recruited 12 participants, 6 females and 6 males, with an age range from 18 to 58 (mean $\bar{x} = 35.83$, median $med(x) = 35$ and $SD = 13.74$).

Gender	Female	50.00 %
	Male	50.00 %
Age	18-25	16.67 %
	26-40	50.00 %
	41-55	16.67 %
	56 or older	16.67 %
Job	Student	25 %
	Executive	0 %
	Employee	75 %
	Self-employed	0 %
	Retired	0 %
	Unemployed	0 %
	Other	0 %
Highest level of education	Primary education	25 %
	Secondary education	50 %
	Bachelor's degree	25 %
	Master's degree	0 %
	Doctoral degree	0 %

Table 1: Demographic information for participants from our elicitation study.

Participants are asked to rate from 1 to 7 (1=not agree and 7=agree), if they used a computer, smartphone, a tablet, a game console or a Kinect frequently :

- Uses a computer frequently: $\bar{x} = 3.58$
- Uses a smartphone frequently: $\bar{x} = 6.67$

- Uses a tablet frequently: $\bar{x} = 3.67$
- Uses a game console frequently: $\bar{x} = 1.92$
- Uses a Kinect frequently: $\bar{x} = 1$

From the mean rates, we can notice that the participants are not used to interact with mid-air gestures. They are used interacting with interfaces and technologies such as WIMP (windows, icons, menus, and pointing) and surface gestures. None of the participants knows the apparatus (the Leap Motion Controller) used in the experiment.

3.3 Referents

To form the list of common referents, which users may perform frequently during the maintenance operation, we started by looking into previous research, such as [Wobbrock et al., 2009, Piumsomboon et al., 2013]. We used [Piumsomboon et al., 2013] study to categorize the referents chosen and grouped them into six categories. Table 2 lists the final set of referents chosen for the study.

Category	Tasks
Transform	1. Move 2. Enlarge 3. Shrink
Simulation	4. Play/Resume 5. Pause 6. Stop/Reset 7. Mute 8. Volume up 9. Volume down

Category	Tasks
Browsing	10. Previous 11. Next 12. Scroll
Menu	13. Open menu 14. Close Menu
Selection	15. Single selection
Collaboration	16. Start call 17. End call

Table 2: The list of the tasks used in the study.

3.4 Procedure

At the start of each session, participants were asked to fill out a consent form and demographic questionnaire, where we ask about prior experience with related devices. Participants were informed about the study purpose and briefed with an introduction to mixed reality and the device.

Participants were presented with the list of referents and asked to design two gestures for each presented referent, followed by the identification of their preferred gesture (symbol) for the referent. Referents were presented in a random order to the participants. They were encouraged to try out several possibilities for each referent, and they could revisit their gestures at any time. Participants were asked to propose two gestures to reduce legacy bias, [Morris et al., 2014] production technique.

They were asked to rate the gesture from one to ten in terms of goodness (how well the proposed gesture fits the referent), having one as very poor fit and ten excellent fit. They were also asked to rate the gesture from 1 to 5 in terms of complexity (1=most complex 5= simplest gesture), memorability (1=hardest to remember and 5=easiest to remember) and fun (1=is the least funny 5=is the funniest).
A gesture sheet (per participant) was filled with the thinking time (time in seconds), time between the moment the referent was presented and the moment the participant proposed the gesture, for each gesture.

After participants completed each of the referents, they are asked to complete an IBM CSUQ (Computer System Usability Questionnaire) questionnaire, to elicit their feedback about the experience. They were asked to rate 16 questions from one to seven in term agreement (1=do not agree and 7=agree). The result rates are shown in Table 4.

3.5 Results

With 12 participants, 17 referents, a total of 106 different gestures were made. After choosing the preferred ones, we got 51 unique gestures.

3.5.1 Classification of gestures

The classification of gestures was made with the six dimensional taxonomy of [Piumsomboon et al., 2013] (explained in Section 2.9) and only on the preferred ones. As shown in Figure 23, the most common characteristics of gestures were *static pose and path* (49%), *metaphorical* (33%), *object-centric* (37%), *discrete* (61%), *dominant unimanual* (78%), and *in-the-air* (100%).

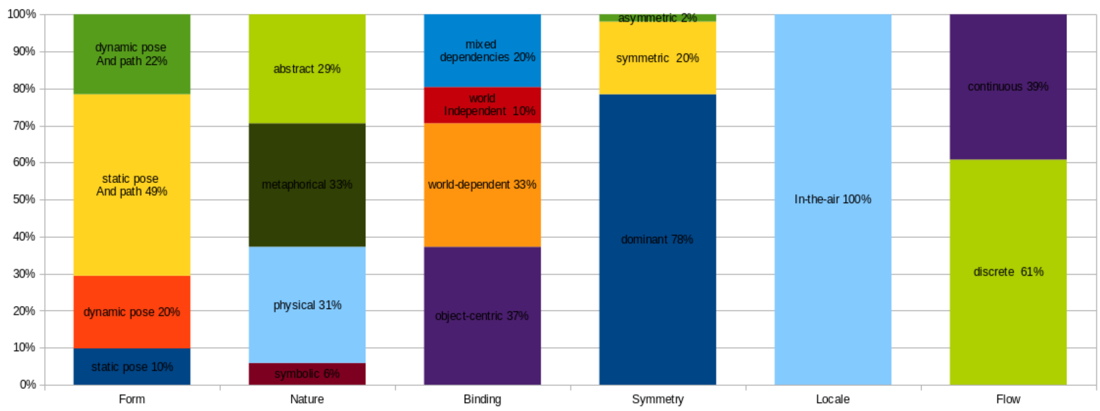


Figure 23: The proportion of gestures in each category of the dimensions.

In the *form* dimension, we calculated it for each category. As shown in Figure 24, for each category of *form*, gestures were made by the *dominant* hand, with a high disparity (between 64% and 100%). Those results were not surprising since most participants use smartphones and tablets on a daily basis.

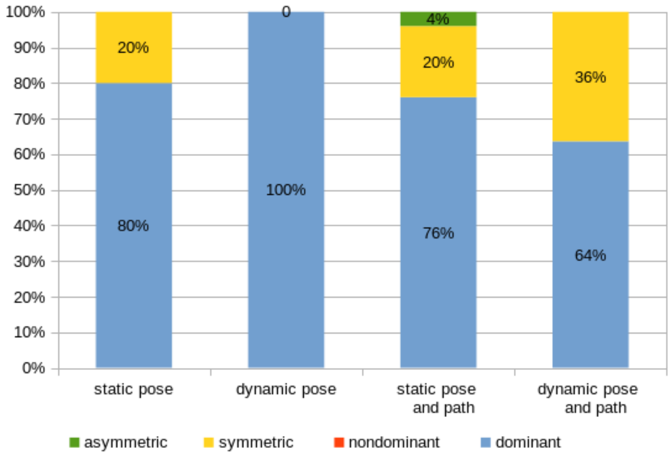
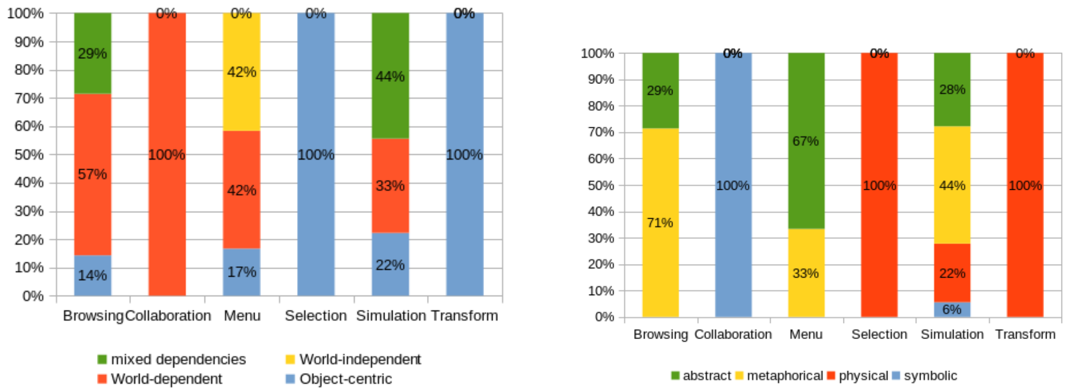


Figure 24: The proportion of gestures in each category of the dimensions *Form* per category of the dimension *Symmetry*.

In the *binding* dimension, overall the gestures were dominantly *object-centric* (37%) and *world-dependent* (33%). The Figure 25a shows a more detailed view of the dimension *binding*. The gestures chosen to perform *transform* (100%) and *selection* (100%) tasks were entirely *object-centric*. The *collaboration* gestures were entirely *world-dependent*. For *browsing*, the gestures were predominantly *world-dependent* (57%). For *menu*, the gestures were predominantly *world-independent (57%).* (42%) and *world-dependent* (42%). For *simulation*, the gestures were predominantly *mixed dependencies* (44%), followed by *world-dependent* (33%).

Within the *nature* dimension, overall the gestures were dominantly metaphorical (33%), physical (31%) and abstract (29%). The Figure 25b shows a more detailed view of the dimension *nature*. The gestures chosen to perform *transform* (100%) and *selection* (100%) tasks were predominantly *physical*. The *collaboration* gestures were entirely *symbolic*. For *browsing*, the gestures were predominantly *metaphorical* (71%). For *menu*, the gestures were predominantly *abstract* (67%). For *simulation*, the gestures were predominantly *metaphorical* (44%), followed by *abstract* (28%) and *physical* (22%).



(a) The proportion of gestures in each category of the dimensions *binding* per categories of referents.

(b) The proportion of gestures in each category of the dimensions *nature* per categories of referents.

Figure 25: Analysis by referent category.

3.5.2 Agreement rate

For the agreement rate, we used the AGATe (AGreement Analysis Toolkit) to compute it. The AGATe used the agreement rate proposed by [Vatavu and Wobbrock, 2015], which is a less optimistically version of [Wobbrock et al., 2009] formula. The agreement rate (AR) is the number of pairs of participants in agreement with each other divided by the total number of pairs of participants that could be in agreement (Figure 26).

$$\mathcal{AR}(r) = \frac{\sum\limits_{P_i \subseteq P} \frac{1}{2}|P_i|\left(|P_i| - 1\right)}{\frac{1}{2}|P|\left(|P| - 1\right)}$$

Figure 26: Agreement rate [Vatavu and Wobbrock, 2015]

The agreement rates for all 17 referents are shown in Figure 28. While there is low agreement in the gesture set for referents such as *open/close menu* there were notable gestures that stood out with higher scores.

Agreement rates ranged from 0.015 to 1, and AR is classified as low for a rate $\leq .100$, medium when between $.100 - .300$, high when between $.300 - .500$ and very high when $> .500$. The AR interpretation is shown in Figure 28. As we can see, the AR was *high* or more for 58.82%.

3.5.3 Criteria

In addition, the qualitative data given during the experiment are also included in the analysis of different criteria, shown in Table 3.

Time The *thinking time* (time between the moment the referent was presented and the moment when the participant knows what gesture to propose) was recorded for each gesture proposed by the participant.

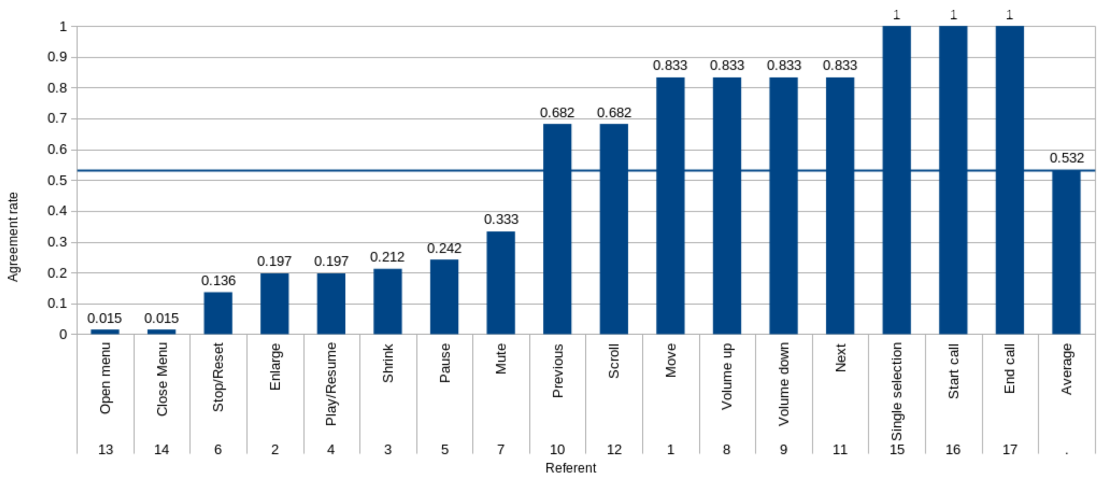


Figure 27: Agreement rate for the 17 referents in ascending order, with a confidence interval of 95% .

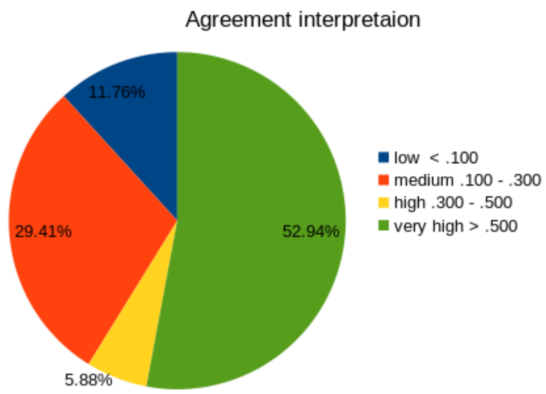


Figure 28: Agreement rate interpretation.

The gesture who got the most thinking time was *Openmenu* ($M = 58.75$ $SD = 44.39$). In the other hand, *Closemenu* got a much better score ($M = 6.58$ $SD = 6.08$) because most of the time, participants used the same gesture as for *Openmenu* . Same thing happens for *Play/Resume* and *Pause*.

When participants found it difficult to come up with a gesture, they would often resort to using metaphors from familiar UI. For the referents with a mean *thinking time* < 10, such as *Enlarge*, *Shrink* or *Scroll*, the lower score can be explained by the fact participants rely on existing gestures they already use daily (with computers and smartphones/tablets).

Complexity The participants rated their effort in performing the gesture. The execution of gestures were rather positive, with the most complex gesture got a mean of 4 (maximum of 5=simplest gesture). And we got the highest $SD = 0.85$. The participants have not encountered significant difficulties performing the gestures. From the scores, all the gestures performed were simple.

Memorability We asked whether the gestures performed were memorable or not, with no extra help to recall them. The less memorable gesture got a mean of 4 (maximum of 5=easiest to remember). And we got the highest $SD = 0.74$. From the scores, all the gestures performed were easy to remember.

Fun We asked whether the gestures performed were fun or not. The less fun gesture got a mean of 2.25 (maximum of 5=is the funniest). And we got the highest $SD = 1.71$. Gestures like *Single selection*, *Play/Resume*, *Pause* got a low score under 2.5. The participants explained that low scores were due to boring gestures. Gestures they usually already use.

Good match Gestures proposal seem to have a good match. The worse match gesture, got a mean of 7.92 (maximum of 10=excellent fit). And we got the highest $SD = 0.90$. From the scores, all the gestures were a good fit.

	Referent	Time		Complexity		Memorability		Fun		Good match	
		M	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
1	Move	5.92	3.55	5.00	0.00	4.92	0.29	4.08	0.79	9.50	0.52
2	Enlarge	7.42	4.42	4.83	0.39	4.83	0.39	3.75	0.75	8.83	0.58
3	Shrink	4.83	3.74	4.83	0.39	4.92	0.29	3.67	0.89	8.83	0.72
4	Play/Resume	17.42	11.39	4.83	0.58	4.17	0.39	2.33	1.07	8.00	0.74
5	Pause	7.58	9.32	4.92	0.29	4.33	0.65	2.75	0.97	7.92	0.79
6	Stop/Reset	26.67	22.94	4.83	0.39	4.08	0.67	3.08	0.67	8.00	0.60
7	Mute	39.08	27.64	4.33	0.49	4.00	0.74	3.58	0.79	7.92	0.90
8	Volume up	9.00	7.83	4.83	0.39	4.92	0.29	3.42	0.79	9.00	0.60
9	Volume down	6.50	5.82	4.92	0.29	4.83	0.39	3.50	0.67	9.08	0.51
10	Previous	7.00	7.68	5.00	0.00	5.00	0.00	3.58	0.67	8.92	0.67
11	Next	4.33	5.19	5.00	0.00	4.92	0.29	3.58	0.67	8.92	0.67
12	Scroll	2.83	1.59	4.67	0.00	4.92	0.29	3.33	0.65	9.08	0.79
13	Open menu	58.75	44.39	4.58	0.49	4.17	0.58	4.00	1.71	8.08	0.51
14	Close Menu	6.58	6.08	5.00	0.51	4.17	0.39	3.67	0.89	8.08	0.51
15	Single selection	4.33	4.50	5.00	0.00	5.00	0.00	2.25	0.87	9.67	0.65
16	Start call	12.75	31.05	4.00	0.74	5.00	0.00	4.83	0.39	9.50	0.52
17	End call	7.58	8.13	4.00	0.85	5.00	0.00	4.75	0.45	9.58	0.51

Table 3: Result of the subjective measures questionnaire (time in seconds).

3.5.4 CSUQ

The CSUQ suggests that the global satisfaction of the participants involved in the experiment follows a positive trend, as seen in Figure 29. Most of the rates are greater than 5. For Q7 to Q9 some participants were confused and preferred answer N/A.

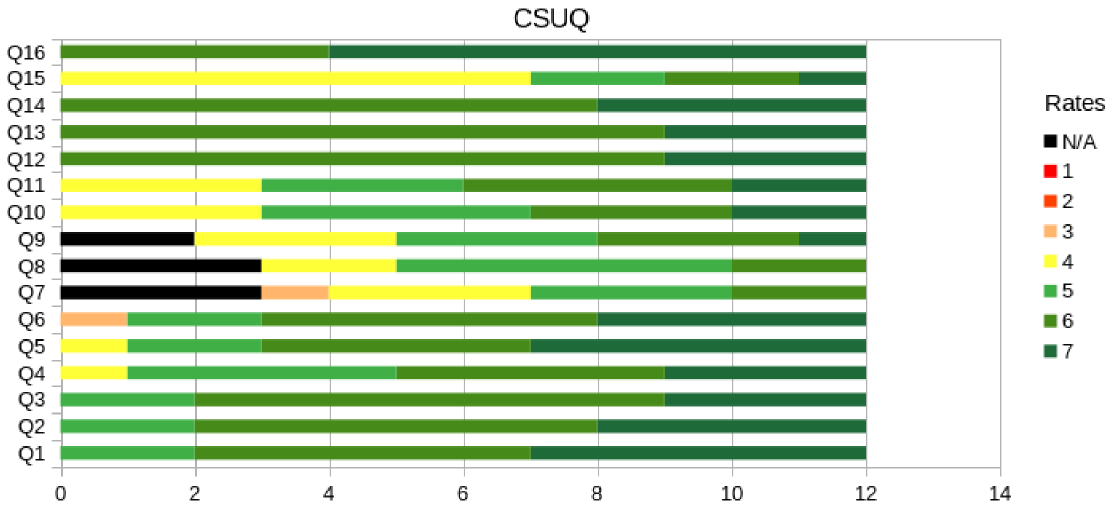


Figure 29: Results from the IBM CSUQ questionnaire.

The CSUQ rates for all 16 questions are shown in Table 4. The question n° 16 confirms the positive trend since we obtained $M = 6.67$, $MD = 7.00$ and $SD = 0.49$, which is the most positively assessed question.

Questions 7 to 9 got the most negatively assessed questions (Q7 : $M = 3.50$, $MD = 4.00$ and $SD = 2.34$). Some participants were troubled by the questions. The most heard comment was about the system, since it does not really exist and the participants needed to visualize it. Some had trouble with that abstraction.

The second most positively assessed question was Q14 ($M = 6.33$, $MD = 6.00$ and $SD = 0.49$), which demonstrates that participants liked using the system (some comments were about the new type of interaction, unknown for most of the participants but appreciated).

	Question	Mean	MD	SD
1	I am happy with the ease of use of this system	6.25	6.00	0.75
2	This system is easy to use	6.17	6.00	0.72
3	I can effectively accomplish my task with this system	6.08	6.00	0.67
4	I am comfortable with this system	5.75	6.00	0.97
5	The system is easy to learn	6.08	6.00	1.00
6	I think I can quickly become productive using this system	5.92	6.00	1.16
7	The system gave me clear error messages to troubleshoot	3.50	4.00	2.28
8	I was able to correct each error simply and quickly	3.75	5.00	2.34
9	Information provided by the system, such as online help, messages, documentation, was clearly provided by the system	4.33	5.00	2.23
10	I easily found the information I needed	5.33	5.00	1.07
11	The information was useful for me to carry out the tasks and scenarios	5.42	5.50	1.08
12	The information was clearly presented	6.25	6.00	0.45
13	The system interface was nice	6.25	6.00	0.45
14	I liked using the system interface	6.33	6.00	0.49
15	The system had all the features I wanted	4.75	4.00	1.06
16	Overall, I am satisfied with the system	6.67	7.00	0.49

Table 4: IBM CSUQ questionnaire: M, MD and SD.

3.5.5 Consensus Gesture Set

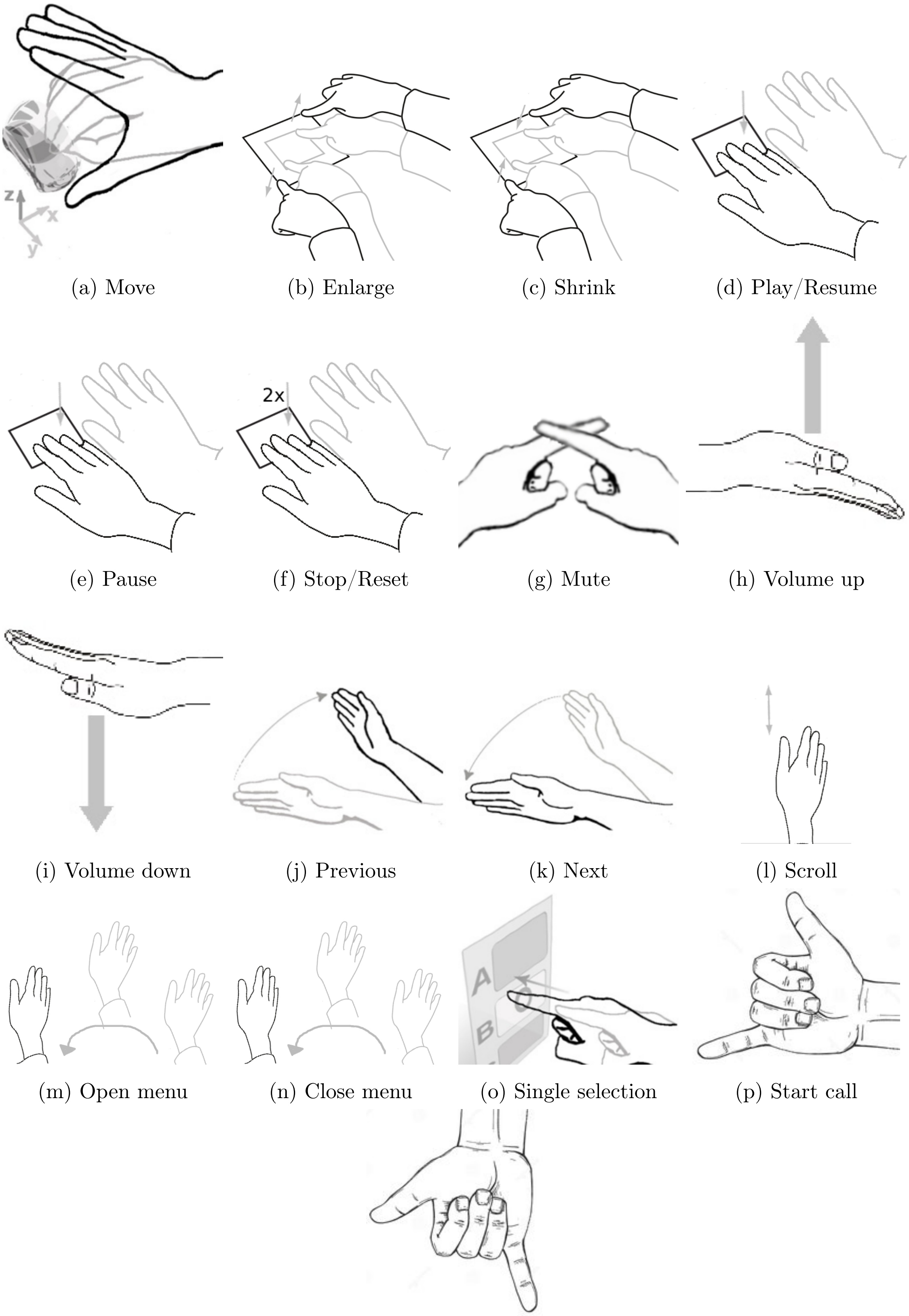
The Table 5 describes each gesture from the consensus gestures set for all 17 referents.

	Referent	Description of the elicited gesture
1	Move	Grab the object and move it.
2	Enlarge	Point the object with 2 index and separate them (scale).
3	Shrink	Point the object with 2 index and join them (scale).
4	Play/Resume	Touch the object with hand. Hand is flat.
5	Pause	Touch the object with hand. Hand is flat.
6	Stop/Reset	Touch the object with hand 2 times. Hand is flat.
7	Mute	Make a cross with the 2 index fingers. Fists forward.
8	Volume up	Palm faced up and raise hand.
9	Volume down	Palm faced down and lower hand.
10	Previous	Flat faced left and swipe left.
11	Next	Flat faced left and swipe right.
12	Scroll	Palm faced down and swipe down/up
13	Open menu	Palm faced forward, make a half-circle right to left (180° - 0).
14	Close Menu	Palm faced forward, make a half-circle right to left (180° - 0).
15	Single selection	Touch with index.
16	Start call	Make a vertical fist and open the thumb and little finger (make a phone symbol).
17	End call	Make a vertical fist and open the thumb and little finger and turn it -90° (phone symbol with hand and hang up).

Table 5: Consensus gesture set : description of the elicited gestures.

As we can see in Figure 30 the original gesture set (51) was reduced to 15 unique gestures. We allowed participants to assign the same gesture to different referents. This set is smaller than the number of referents, which means that some gestures have more than one referent. It is the case for referents *Play/Resume* and *Pause*, which have the same gesture. The same applies to the referents *Open menu* and *Close menu*. Even having two gestures per referent in the same category, there was no conflict. For those referents, the gestures acted like a reverse.

Reversible and reusable gestures were also present in the consensus set. Reversible gestures are defined as those when performed in an opposite direction render opposite effects e.g. *enlarge*, *shrink*, *next* and *previous*. For increase/decrease *volume* there is a slight difference in the gesture, one flat up and other flat down. Reusable gestures are defined as those that were used commonly for different tasks, but participants felt had common attributes e.g. *play/pause* and *open menu/close menu*.



(a) Move (b) Enlarge (c) Shrink (d) Play/Resume

(e) Pause (f) Stop/Reset (g) Mute (h) Volume up

(i) Volume down (j) Previous (k) Next (l) Scroll

(m) Open menu (n) Close menu (o) Single selection (p) Start call

(q) End call

Figure 30: Consensus Gesture Set

3.6 Limitations

The number of participants who completed the elicitation procedure was too small, they were all adults and not used to different interaction systems. Undoubtedly, children, would behave differently, especially because they relate less to WIMP interactions.

Another limitation is the lack of maintenance experts among the participants. Due to their experience, the gestures would be more adapt to field work. e.g. knowing that during the maintenance the user need to have all the time a tool in his hand, will change all the gestures proposed.

As mentioned in previous studies like [Aigner et al., 2012], a potential limitation is possible dependence on culture. Participants should come from different countries and cultures. For the referent *stop*, a central-European frequently does it by facing the palm of a single hand towards the object, while the Japanese equivalent is crossing both arms.

In this study, referents were presented in a random order, and we asked the participants to propose 2 gestures for each referent to reduce legacy bias, but they were given one after the other, which introduced a bias on the thinking time. For the first gesture proposed, the thinking time is an exploitable data, but not the second. Knowing the process, participants could start thinking to the new one straight ahead and that time not being recorded.

An important step would be to validate the consensus gesture set with new participants.

3.7 Mapping gestures to context

Concerning the application, it will be divided in several modes, such *selection*, *instruction*, *video* and *collaboration*.

During the *selection* mode, the user will only interact with a menu to select the printer (brand and model) and the maintenance task to execute. The menu will be fixed to the environment and the user will be able only to select and scroll (if necessary), using gestures Figure 30o and Figure 30l.

During the *instruction* mode, the user will have a window object where the information about the maintenance instruction is displayed and the 3D objects (arrows in our case) to show where the instruction should be applied. *Video* and *collaboration* modes can be activated in this mode. The gestures for those 3 modes have *mixed dependencies*. The gestures used in this mode are Figure 30o to select the *video* or the *collaboration* mode and Figure 30k and Figure 30j to browse the instructions (previous and next). User can use Figure 30a, Figure 30b and Figure 30c to *transform* (respectively grab, enlarge and shrink) the window object where the instructions are displayed.

During *video* mode, which is just a video player with the maintenance instruction video, user can use the *simulation* gestures, Figure 30d, Figure 30f, Figure 30i, Figure 30h and Figure 30g. User can use the *transform* gestures with the video player.

In *collaboration* mode, user can use Figure 30q and Figure 30p, to launch and end the call and the *transform* gestures to manipulate the call screen.

The *open/close menu* gestures, Figure 30m and Figure 30n, can be used at any moment and anywhere *world-independent*.

4 Case study : printer maintenance application

4.1 Motivation

The set of guidelines proposed in the previous sections provide an insight to design a MR printer maintenance application.

Whether a printer is out of ink, missing paper or the paper is jammed, printer troubleshooting can be a pain. And the only support is a manual or videos. Now, with mixed reality, printer repair and support can be easier. This could be a company offering maintenance service, sending a technician to the client workplace. Or a printer brand not offering maintenance support in certain countries and sending a mixed reality support to help the client.

This work aims to propose a system of mixed reality maintenance on printers, using a Leap Motion Controller and a computer. The system uses mixed reality to assist the technician/user doing the different maintenance tasks. The tasks are decomposed in several instructions that are displayed one by one, using 3D virtual objects.

This work aims to propose also, an alternative to WIMP interactions using the consensus set gestures. Using gestures to launch some actions, can lighten the UI.

4.2 Analysis and prototypes

The user selects from a list (Figure 32), his printer model and the maintenance task to achieve (can be a code error from the printer), and the system displays the instructions related to the maintenance task. This can be improved by scanning the QR code for the printer model and/or the maintenance task. The instructions can be text, images, videos or 3D objects. The user can call an engineer of the printer brand for remote assistant. The advantage is that the engineer can see the same thing as the user. The user has more support, he has a route to guide him through any problems, issues or advice. The user is guided through processes step-by-step through as live simulation. The instructions move with the user, pointing them to the tools and to the parts they need and show where they need to apply them. Another advantage, is that all steps can be validated one by one, and we can generate a report for the manager and/or client.

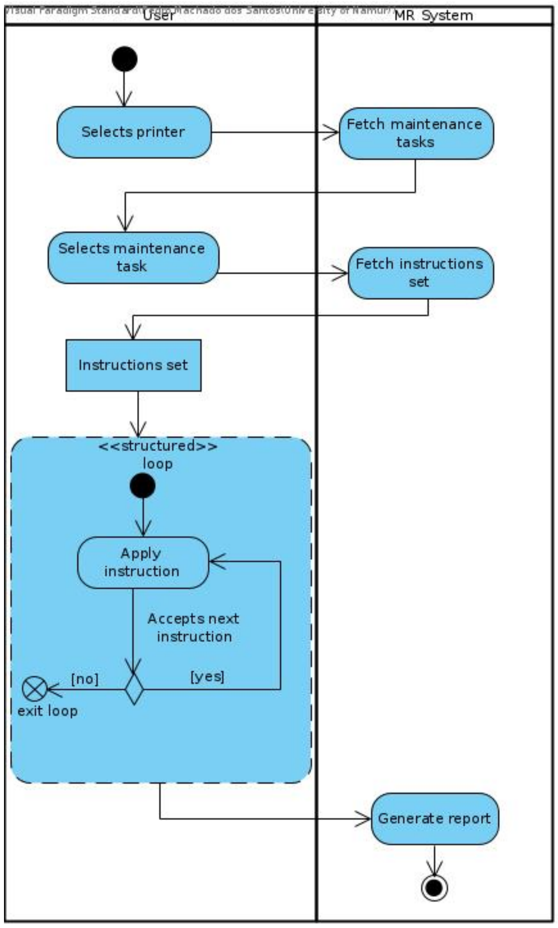


Figure 31: User process

4.2.1 Printers

The first step of the maintenance is to identify the printer. For that, the user selects the brand, category and model. As said before, this can be improved by scanning the QR code for the printer model. Some brands and models have already proposed this feature. We have identified several categories (partial list) such as ink jet printers, laser printer, dot matrix printer, dye-sublimation, LED Printers and 3D printers. Depending on the printer categories, the maintenance task can change, and be completely different. For example, replacing the ink on an ink jet printer and a laser one, are not exactly the same, the laser printer has more steps.

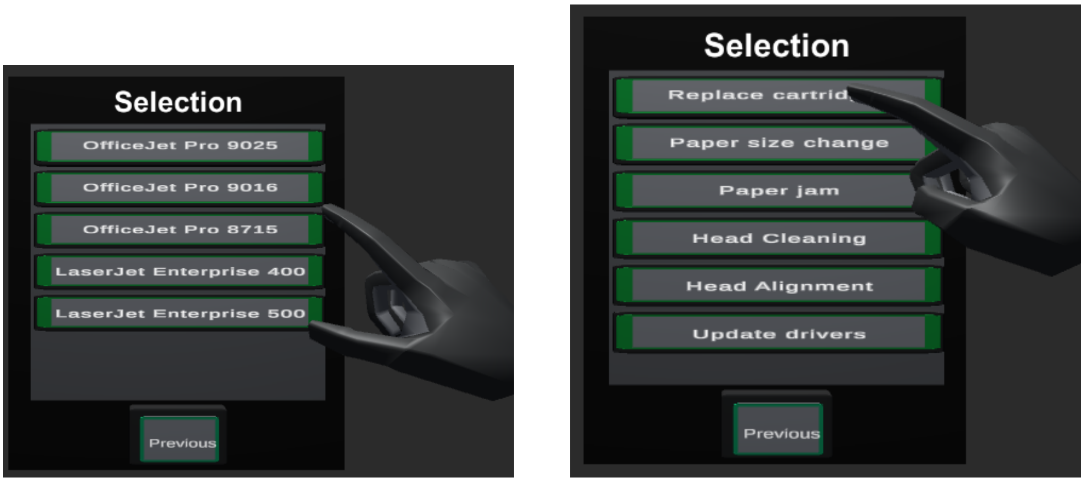


Figure 32: Printer model and task selection.

4.2.2 Maintenance tasks

All the maintenance tasks can be categorized into the different types of maintenance, such as reactive, preventive, productive, predictive. Within the project, we selected some common maintenance tasks, but did not categorize them. We do not focus on maintenance, but on the gestures. However, if the focus was to build a quality maintenance plan, it was necessary to set the level of control and understand all aspects of the business requiring maintenance. In that case, it would be interesting to categorize the maintenance tasks. To go further, all data from all the maintenance interventions could be collected and used for predictive maintenance.

The second step is to select the maintenance task. A list of tasks is proposed to the user. That list is based on the printer model. The following Table 6 shows the maintenance tasks created in mixed reality for this project, but others can be added. For some software maintenance tasks, the user will be asked to choose his operating system (OS).

Maintenance tasks	
Replace toner cartridges	
Paper size change	Hardware
Paper jam	
Head Cleaning	
Head Alignment	
Update drivers	Software
Update firmware	
System reset	
Remote assistance	

Table 6: Maintenance tasks.

4.2.3 Instructions

The user will get the set of instructions given the printer model and the task to accomplish. Each instruction is composed of :

- an order,
- a message,
- a video (optional),
- a virtual object.

The instruction order is necessary because the instructions need to be executed in a specific order. In the future, we could add (to detail more) or remove (unnecessary) an instruction from the task, keeping an order field is an advantage. The message is just the textual instruction. The video, which is optional, would be an additional help, just in case it is a complex instruction. It could be the whole task video or only specific to the instruction. The virtual object, is here to indicate where the instruction should be applied. In this study case, we only used two types of 3D arrows (Figure 33).

Figure 33: Virtual 3D objects : arrows.

Other types of objects could be used, e.g. printer parts could be modeled in 3D objects as in Figure 34a or a more simple solution just en-light the part with a flat object as in Figure 34b.

(a) 3D approach[1]

(b) Flat approach, [Novikov, 2019]

Figure 34: Virtual objects examples.

[1]https://www.ptc.com/en/blogs/service/augmented-reality-maintenance-and-repair

The arrows are placed to the relative position of the printer part. That way, when creating the maintenance task there is no need to worry about the environment.

Figure 35: Example of an instruction, composed of a panel and virtual objects.

4.3 Architecture

Independently of any specific technology, the top-level architecture of our printer maintenance application is illustrated in Figure 36. It is a three-tier architecture, with a MVC (model–view–controller) pattern for the presentation layer.

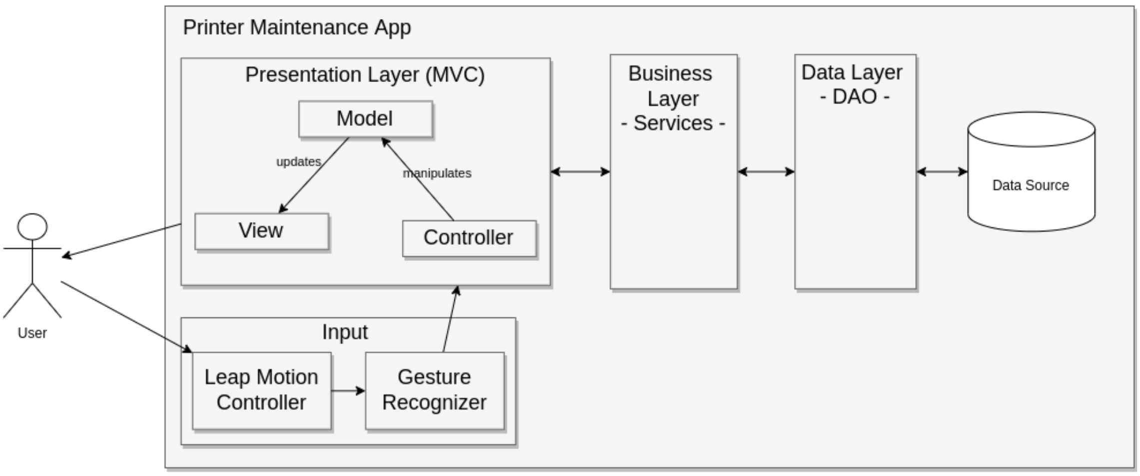


Figure 36: Top level architecture.

As we see, the first source of information comes from the user. The user makes a gesture and the *Leap Motion Controller* captures the movements of user hands (this information is constantly tracked). It captures information such as position, orientation and velocity from the arms, hands and fingers.

The *Gesture Recognizer* uses the hands data provided by the *Leap Motion Controller* and tries to identify our custom gestures. Allows the application to manage the different supported gestures. It allows application components to define a set of actions to be executed when a given gesture is recognized.

The *Presentation Layer* is a graphical representation schematized under the MVC model. It receives a request from the client (through the *Gesture Recognizer*) and redirects it to the controller. The request is processed by the controller, which returns the response to the client. The controller defines a set of actions to be executed when a given gesture is recognized.

The *Business Layer* will take care of the management of all needs collected during analysis, such as get the all available printers, the maintenance tasks related to a given printer or the instructions given a printer and task.

The *Data Layer* manages the connections with the database and translates the requests received from the *business layer* into requests that can be interpreted by *data source*. It allows the access to the data present in the *data source*.

The *Data source* is where the data are stored. Can be the mechanism responsible for recovering and saving data (i.e. database).

4.4 Technology

The two main technologies used in this study case are the *Leap Motion Controller (LMC)* and *Unity*. The development was done on a Windows 10 and Unity 2021.3 environment due to the *LMC* system requirements.

4.4.1 Unity

Unity is a cross-platform 2D/3D game engine, it is a platform for content creation. Unity administers things like 3D rendering, physics, and collision detection. It uses C# to handle code and logic.

Unity defines some important elements, which are the key pieces to build any Unity application. A brief description of each of them is listed below:

Assets : are the representation of project item(s). It can be imported files (supported by Unity) like an 3D model, audio and sound file, images, textures or any of the other file types.

GameObject : every object present is a GameObject. It acts as holder for components like the Transform, Light, Script, and RigidBody components.

Components : are the basic building blocks of objects and their activities in a game. They act as functional pieces for each GameObject. By default, every GameObjects have a Transform Component set automatically because, it dictates the position, rotation and scale within the environment.

Scripts : are behaviors components that can be applied to GameObjects. They tell GameObjects how to behave. They are written in C# language.

Scenes : are the base or the parent object, where the GameObjects are placed. One or more scenes can be added to the application. They are linked together and will be loaded in terms of specific objectives.

Prefab : are reusable GameObject components. Prefabs can be added into multiples scenes, as many times as necessary per scene.

4.4.2 Leap Motion Controller

The LMC (subsubsection 2.4.2) captures (with hand tracking software) all the subtlety and complexity of natural hand movements. The study case uses the *Gemini* software version, which was the latest version available. This version only proposes 3 API : (*Unity*, *Unreal* and *LeapC*).

The *Unity API* was selected for the development. A *Unity* plugin is available and provides the tools and utilities to connect *Unity* applications to hand tracking data. The main features are :

- *Leap Service Provider* : it handles communication with the hand tracking hardware and provides Frame objects containing Leap hands. It provides all hand tracking data and ensures that the virtual hands have the correct transform applied.
- *Capsule Hands* : generate a set of spheres and cylinders to render the user hands using hand data provided by the LMC. It allows hand visuals to scale to user hand size and is a reliable way to visualize the raw tracking data.
- *Hand Models* : used to visualize hand tracking data using custom hands style's (a variety styles are available, such as skeleton, outline or ghost hands).

The LMC can be used in 3 different tracking orientations:

- *Desktop* : enable the tracking camera to be placed on a flat surface. This mode assumes that the hands are parallel to the ground, palm down, as the camera is looking straight up.
- *Screen-top* : enable the tracking camera to be mounted above a screen (facing down towards the user).
- *Head-Mounted Display* : enable the tracking camera to be mounted above a HMD (fastened to a VR/AR headset).

This setting determines what and how the Leap Motion controller is expecting to see. Selecting the most suitable mode is important, otherwise the perspective is different.

4.5 Development

4.5.1 UI

As previously explained, the user interface of the application is handled by the *Scenes*, where *GameObjects* are added.

Several *GameObjects* were created to represent the 3D objects of the UI, such as :

Selection panel : displays a selection menu, where the user can select the printer brand, model and the maintenance task (Figure 32). It is a reusable *GameObject*. A *script* controller was added to load the data through a service.

Instruction panel : displays each instruction (the message and the arrows to indicate where to apply it) one by one (Figure 35). A controller was added to load the data through a service.

Video player panel : displays and plays a video on an 3D object (Figure 37a).

Video caller panel : displays a communication feed (Figure 37b).

(a) Video Player. (b) Video Caller.

Figure 37: Video Player and Caller UI.

The *Gesture Recognizer* was added to the previous *GameObjects*. Other *GameObjects* and *Prefabs* related to the LMC were added too, like the *Serice Provider* and the *Hand Models*.

4.5.2 Gestures interaction

As mentioned in Section 3.5, some gestures are *object-centric*. This means, gestures are defined within respect to the objects being manipulated (i.e. *move*, *select* or *enlarge*). For those types of gestures, we need to define the interaction space around the object, Figure 38 shows that interaction space, the green borders boxes. This way, any gesture executed inside the interaction space, is relative to that only object.
In this study, we need to be able to have multiple objects displayed at the same time, e.g. the instruction and the video call windows, and be able to interact with them separately. Doing a gesture in the interaction space of one window, will not have any effect on other one (e.g. grabbing an object to move it, will not influence on the other objects).

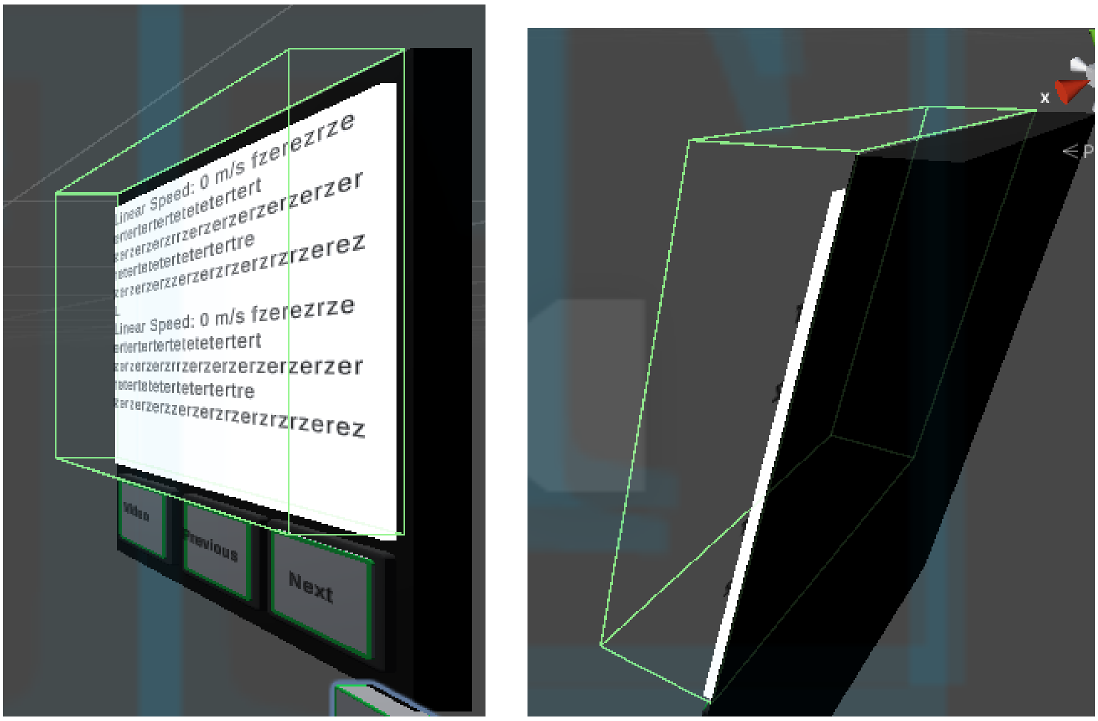
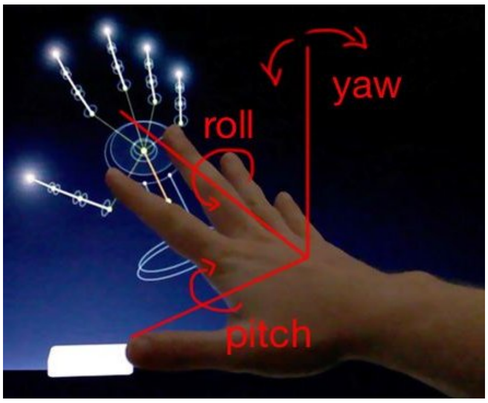


Figure 38: Gesture interaction area.

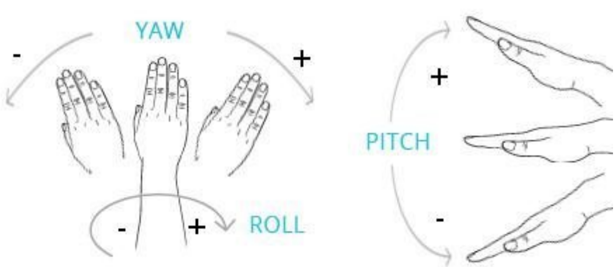
4.5.3 Gestures recognition

For recognizing gestures, we need to recognize certain hand poses (e.g. the index finger, a flat palm, a fist, the phone symbol, *etc.*). The most important aspects for recognizing poses are distances and angles.

When recognizing poses, we need to acquire the hand orientation (i.e. hand pointing forward or upward). For this, we need to get 3 angles (Figure 39), the pitch (angle around the x-axis), yaw (angle around the y-axis), and roll (angle around the z-axis).



(a) [Michael Kipp,]



(b) [Suarez Fernandez et al., 2016]

Figure 39: Leap Motion hand orientation: the pitch (angle around the x-axis), yaw (angle around the y-axis), and roll (angle around the z-axis).

The Listing 1 shows the code for recognizing the pose for a swipe. The swipe pose is defined by the hand having a roll (angle around the x-axis) between 0.5 and -2.8 radians (28.65 and -160.43 degrees). Another feature is that the thumb cannot touch another finger (`firstHand.PinchStrength < 0.3f`).

```
private bool isSwipeHorizontal() {
  return firstHand.PalmNormal.Roll<0.5f && firstHand.PalmNormal.Roll>-2.8f
    && firstHand.PalmNormal.Yaw < -0.5f && firstHand.PalmNormal.Yaw>-2.6f
    && firstHand.PinchStrength < 0.3f;
}
```

Listing 1: Code for recognising a swipe (angles are in radians).

For some gestures, we may need to get some information about the fingers, such as the type of finger (i.e. thumb, index, middle, ring or little), the distance between fingers or if it is extended or not. The Listing 2 shows the code for recognizing the pose for the phone symbol. The pose is recognized if the thumb and the little finger are extended (open). The other ones need to be not extended, which means closed. And finally, the palm needs to be vertical, the roll (angle around the z-axis) is tested between two angles ($-1.6f$ and $-3f$ radians, -91.67 and 171.89 degrees).

```
private bool isPhonePose() {
  return firstHand.Fingers[(int)Finger.FingerType.TYPE_THUMB].IsExtended
    && firstHand.Fingers[((int)Finger.FingerType.TYPE_PINKY)].IsExtended
    && !firstHand.Fingers[((int)Finger.FingerType.TYPE_MIDDLE)].IsExtended
    && !firstHand.Fingers[((int)Finger.FingerType.TYPE_INDEX)].IsExtended
    && !firstHand.Fingers[((int)Finger.FingerType.TYPE_RING)].IsExtended
    && firstHand.PalmNormal.Roll < -1.6f
    && firstHand.PalmNormal.Roll > -3f;
}
```

Listing 2: Code for recognising the phone gesture (angles are in radians).

For gestures recognition (i.e. swiping, tapping, waving, *etc.*), we need motion and the path motion recognition for complex gestures. Motion can be detected by measuring speed, velocity (speed and direction) or acceleration. Defining some interaction spaces (subsubsection 4.5.2) may be necessary to avoid triggering events by accident.

The velocity is a vector of movement. In the Leap Motion API, is the rate of change of the palm position.

The Listing 3 show the code for recognizing a left swipe. First, the swipe pose needs to be recognized and afterwards the palm velocity is tested on the x-axis. Having a palm velocity < 0, means it is a left swipe.

```
public bool isLeftSwipe() {
  return isSwipeHorizontal() && firstHand.PalmVelocity.x < 0.15f;
}
```

Listing 3: Code for recognising a left swipe.

The gestures are recognized by the `GestureRecognizer`. It will be updated each time the application is updated, each time there is a new frame (FPS: in average of 200 times/s). To avoid a continuous recognition, the `GestureRecognizer` provides a cool-down time. It is the time it takes to execute the gesture. A default cool-down time is provided, but can be overridden in each gesture. With such a frequency of updating in real-time 3D applications, the process ensures that actions corresponding to the detected gesture are executed only once.

```
public bool isLeftSwipe() {
  if (!foundHands) {return false;}

  SwipePose swipe = new SwipePose(firstHand);
  if (swipe.isSwipeHorizontal()
      && firstHand.PalmVelocity.x < -0.15f
      && currentCooldown <= 0.0f) {
    currentCooldown = swipe.COOLDOWN;
    Debug.Log("Left swipe gesture");
    return true;
  }
  return false;
}
```

Listing 4: Updated code for recognising a left swipe with a cool-down time.

4.5.4 Limitations of the Leap Motion Controller

As mentioned in subsubsection 2.4.2, accuracy drops with higher distances, when the measurements were made outside the interaction zone, but that is not the only limitation. We got some trouble with the recognition of some gestures, like the *start call* gesture. The Figure 40b show the perfect gesture recognition. However, sometimes the Leap Motion Controller had some trouble and we got Figure 40c.

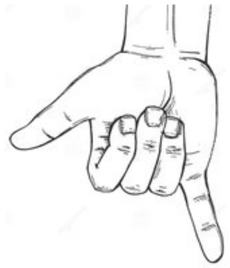
(a) Start call gesture.

(b) Virtual start call gesture, with a good recognition.

(c) Virtual start call gesture, with bad recognition.

Figure 40: Gesture for the referent *start call*

Even worse, the *end call* gesture was badly recognized. When we try to rotate the gesture to start a call (Figure 40b) and transform it in a *end call* gesture, we get Figure 41b. The LMC is incapable of modeling the hand in that position. For the case study, we found an alternative, instead of turning the *start call* gesture down, we turn it up Figure 41c.



(a) End call gesture.

(b) Virtual end call gesture, with a bad recognition.

(c) Virtual start call gesture, alternative.

Figure 41: Gesture for the referent *start call*

5 Conclusion

As described in the introduction of this thesis, the expected outcomes of this work were a gesture elicitation study and a proof of concept application based on the gesture consensus set, in the printer maintenance field.

We first conducted research on the necessary background technology and tools to make a hand gesture elicitation and develop a mixed reality application with the Leap Motion Controller.

Then, we conducted a gesture elicitation study to find the symbols for our list of referents. We asked a group of participants to propose symbols (gestures) for each of ours referents (tasks). With the symbols proposed, we created a consensus set based on agreement, get the most proposed symbol for each referent.

Based on the common symbols chosen by the participants, the consensus set, we developed an application to guide a technician during a maintenance operation. A set of instructions is proposed to the technician after selecting the printer model and the problem to solve.

Nowadays, there are no established universal gesture vocabulary for mid-air interactions with digital content. It is widely acknowledged that gesture identification depends on the context of use, which makes mid-air gestures identification an important design decision. The method of gesture elicitation is applied by designers to help them identify appropriate gesture sets for applications.
[Villarreal-Narvaez et al., 2020] found that the average number of participants in a gesture elicitation study is $M = 25$, and the most frequent choice is 20 participants.

Mixed Reality systems are a promising technology and are here to stay. Due to advances in technologies for gesture recognition, mid-air gestures can be considered the interface of the future for a lot of applications.
It would be interesting to conduct a review of gesture elicitation studies in order to create a universal gesture dictionary. The gestures would be classified, especially by context, so cross-referencing could be made to compare agreement or results could be grouped.

According to [Lin et al., 2019], hand gestures have not been thoroughly evaluated for usability and performance. Fatigue is a known problem, with a dictionary, gestures (e.g. the most elicited ones) could be studied one by one and get a positive or negative feedback. Ergonomic, cognitive and social aspect could be studied too.

In regard to mixed reality health, several studies have been made, such as [Davis et al., 2014] who did a systematic review of cybersickness and [Cometti et al., 2018] who studied the effects of mixed reality head-mounted glasses on cognitive and physiological functions. It would be interesting to conduct studies about the side effects of the gestures and prevent injuries.

List of Figures

List of Tables

A Elicitation documents

UNIVERSITÉ DE NAMUR

UNamur
rue de Bruxelles 61, B-5000 Namur
pedro.machado@student.unamur.be

CONSENTEMENT A PARTICIPER A UNE EXPERIENCE SCIENTIFIQUE

Titre de l'étude: Élicitation de gestes
Professeur: Bruno Dumas
Expérimentateurs: Pedro Machado dos Santos
E-mail de contact: pedro.machado@student.unamur.be

Nous vous remercions de bien vouloir contribuer à cette recherche en effectuant cette expérience. L'expérimentateur vous expliquera le projet pour lequel vous avec accepté de contribuer. Si vous avez une question sur les formulaires ou sur l'explication, veuillez demander à l'expérimentateur avant de débuter l'expérience. Vous pouvez obtenir une copie de ce formulaire de consentement et vous y référer à tout moment.

J'accepte que, si à tout moment je décide de ne plus participer à cette expérience, je peux notifier l'expérimentateur et je serai retiré de l'expérience immédiatement.

Je consens à ce que les données personnelles issues de cette expérience soient traitées à des fins scientifiques. Je comprends que ces informations seront traitées de manière strictement confidentielles et en accord avec la réglementation du respect de la vie privée de l'UNamur.

☐ Veuillez cocher cette case si vous acceptez d'être éventuellement contacté ultérieurement pour d'autres expériences menées à l'UNamur ou pour une interview.
Le cas échéant, adresse e-mail: ……………………………………………….

Déclaration du Participant:
Je, soussigné ……………………………………………………., reconnaît que l'expérience susmentionnée m'a été expliquée avec satisfaction et j'accepte de prendre part à cette expérience. J'ai lu les termes ci-dessus et les formulaires accompagnant et je comprends ce que l'expérience va analyser.

Signature: Date:

Déclaration de l'Expérimentateur:
Je, soussigné Pedro Machado dos Santos, confirme avoir expliqué avec soin la nature, les besoins et, le cas échéant, les risques prévisibles de l'expérience proposée au volontaire.

Signature: Date:

UNamur
rue de Bruxelles 61, B-5000 Namur
pedro.machado@student.unamur.be

QUESTIONNAIRE POUR PARTICIPER A L'EXPERIENCE

Titre de l'étude: Élicitation de gestes
Professeur: Bruno Dumas
Expérimentateurs: Pedro Machado dos Santos
E-mail de contact: pedro.machado@student.unamur.be

Données personnelles (qui seront anonymisées)
Code anonyme: ……… (à attribuer par l'expérimentateur)
Prénom : …………………………. (anonymisées dans la base de données)
Nom de famille : ………………………….
Genre : [ ] Femme [ ] Homme
Age : ……………………. (années)
Profession : [ ] Étudiant [ ] Cadre [ ] Employé
[ ] Indépendant [ ] Pensionné [ ] Chômeur
[ ] Autre : ………….. (préciser s.v.p.)
Domaine d'activité: ……………………………. (p. ex., administration, enseignement)
Études : ……………………………. (domaine de votre dernier diplôme)

J'utilise un ordinateur fréquemment:	Pas d'accord	1	2	3	4	5	6	7	D'accord
J'utilise un smartphone fréquemment:	Pas d'accord	1	2	3	4	5	6	7	D'accord
J'utilise une tablette fréquemment:	Pas d'accord	1	2	3	4	5	6	7	D'accord
J'utilise une console de jeux fréque^{t}.:	Pas d'accord	1	2	3	4	5	6	7	D'accord
J'utilise une Kinect fréquemment :	Pas d'accord	1	2	3	4	5	6	7	D'accord

Données expérimentales

Connaissiez-vous l'appareil utilisé dans l'expérience [ ] Oui [ ] Non
Combien de fois l'utilisez-vous? Rarement 1 2 3 4 5 6 7 Fréquemment

UNamur
rue de Bruxelles 61, B-5000 Namur
pedro.machado@student.unamur.be

FEUILLE D'ENREGISTREMENT DE GESTE: …… DE …… (Date: ……….)

Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):

FEUILLE D'ENREGISTREMENT DE GESTE: …… DE …… (Date: ……….)

Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):
Identifiant du geste :	Description textuelle:	Identifiant du geste :	Description textuelle:
	Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :		Complexité (1..5): Mémorabilité (1..5) : Fun (1..5) :
Temps réflexion :	Score (1..10):	Temps réflexion :	Score (1..10):

UNamur
rue de Bruxelles 61, B-5000 Namur
pedro.machado@student.unamur.be

Questionnaire post-test sur l'utilisabilité d'un système (IBM CSUQ)

Ce questionnaire se veut un moyen d'exprimer votre satisfaction à utiliser un système que vous avez testé. Vos réponses vont nous aider à comprendre quels sont les aspects du système qui posent question pour vous et quels sont les aspects que vous appréciez. Dans la mesure du possible, pensez à toutes les tâches que vous avez réalisées avec le système quand vous répondez aux questions.
Veuillez lire chaque question et indiquer dans quelle mesure vous êtes d'accord ou en désaccord avec la question en l'exprimant suivant une échelle à 7 valeurs allant de 1=totalement en désaccord jusqu'à 7=totalement d'accord. Si vous pensez que la question n'est pas pertinente pour le système que vous avez évalué, cochez la colonne N/A (non applicable). Merci beaucoup pour votre participation!

Identifiant du participant : ……………

Questions		**1**	**2**	**3**	**4**	**5**	**6**	**7**		**N/A**
1. Globalement, je suis satisfait de la facilité d'utilisation de ce système	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
2. Ce système est simple d'emploi	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
3. Je peux effectivement accomplir ma tâche avec ce système	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
4. Je suis à l'aise avec ce système	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
5. Le système est facile à apprendre	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
6. Je pense devenir rapidement productif en utilisant ce système	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
7. Le système m'a fourni des messages d'erreur clairs pour résoudre les problèmes	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
8. J'ai pu corriger chaque erreur simplement et rapidement	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
9. L'information fournie par le système, telle que l'aide en ligne, les messages, la documentation, était clairement fournie par le système	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
10. J'ai trouvé facilement l'information dont j'avais besoin	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
11. L'information m'a été utile pour réaliser les tâches et scénarios	totalement en désaccord	○	○	○	○	○	○	○	totalement d'accord	○
12. L'information était	totalement	○	○	○	○	○	○	○	totalement	○

clairement présentée	en désaccord							d'accord	
13. L'interface du système était agréable	totalement en désaccord	◯	◯	◯	◯	◯	◯	◯ totalement d'accord	◯
14. J'ai aimé utiliser l'interface du système	totalement en désaccord	◯	◯	◯	◯	◯	◯	◯ totalement d'accord	◯
15. Le système disposait de toutes les fonctions que je souhaitais	totalement en désaccord	◯	◯	◯	◯	◯	◯	◯ totalement d'accord	◯
16. Globalement, je suis satisfait du système	totalement en désaccord	◯	◯	◯	◯	◯	◯	◯ totalement d'accord	◯

Lister trois aspects négatifs du système que vous avez regretté:

1. ……………………………..
2. ……………………………..
3. ……………………………..

Lister trois aspects positifs du système que vous avez apprécié:

1. ……………………………..
2. ……………………………..
3. ……………………………..

B Gesture Elicitation Results

1 Preferred Gestures

	Referent	u1						u2						u3						u4						u5						u		
		ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C
1	Move	2	1	5	5	5	9	1	5	5	5	5	10	1	11	5	5	4	10	1	8	5	5	4	9	1	2	5	5	4	9	1	12	5
2	Enlarge	4	2	4	5	4	8	3	6	5	5	4	9	55	12	4	5	5	9	35	7	5	4	3	8	55	11	5	4	3	8	67	15	5
3	Shrink	6	1	4	5	4	8	1	1	5	5	4	9	56	7	4	5	5	9	5	4	5	4	2	7	56	6	5	5	3	9	71	6	5
4	Play/Resume	8	26	3	4	4	6	37	9	5	5	3	9	60	4	5	4	3	8	49	12	5	4	3	8	49	21	5	4	3	8	49	4	5
5	Pause	9	31	4	4	4	6	37	3	5	5	4	9	60	2	5	3	2	8	49	3	5	4	3	8	49	8	5	4	4	7	49	11	5
6	Stop/Reset	11	18	4	5	3	7	39	11	4	4	3	8	62	2	5	3	3	8	60	3	5	4	3	7	60	17	5	4	3	8	62	32	5
7	Mute	13	15	5	4	4	8	41	15	4	3	4	8	64	8	5	3	2	6	73	115	4	3	3	7	41	48	4	4	3	7	64	34	5
8	Volume up	15	3	5	5	4	9	15	7	5	5	4	9	15	2	5	4	4	8	15	2	5	5	3	9	15	12	4	5	3	8	15	15	5
9	Volume down	17	4	5	5	4	9	17	8	5	4	4	8	17	2	5	5	3	9	17	2	5	4	3	9	17	12	5	5	3	9	17	12	5
10	Previous	19	26	5	5	4	8	22	7	5	5	4	8	22	3	5	5	4	9	22	3	5	5	3	8	44	4	5	5	3	9	22	3	5
11	Next	22	19	5	4	3	8	46	2	5	5	4	8	46	1	5	5	3	9	46	1	5	5	4	8	46	7	5	5	4	9	46	6	5
12	Scroll	24	3	5	5	4	8	48	6	5	5	4	8	48	1	5	5	3	9	48	1	5	5	3	8	48	3	5	5	3	10	48	5	5
13	Open menu	26	4	5	4	3	7	52	21	4	4	4	8	37	13	5	4	4	8	69	53	4	3	9	8	76	91	5	4	3	8	51	121	5
14	Close Menu	27	13	4	4	5	7	53	8	4	4	4	8	37	4	5	4	4	8	69	18	4	4	2	8	76	17	5	4	3	8	51	6	5
15	Single selection	29	18	5	5	2	8	29	3	5	5	2	10	29	4	5	5	1	10	29	2	5	5	1	9	29	5	5	5	2	10	29	2	5
16	Start call	31	4	5	5	5	9	31	5	5	5	5	9	31	4	4	5	5	9	31	3	3	5	5	9	31	11	4	5	4	9	31	5	5
17	End call	33	5	5	5	4	9	33	7	5	5	5	9	33	6	5	5	5	9	33	3	3	5	4	9	33	13	4	5	5	10	33	3	5

gesture ID	ID
time in seconds	T
complexity	C
memorability	M
fun	F
goodness	S

6			u7						u8						u9						u10						u11						u12					
M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G	ID	T	C	M	F	G
5	4	10	1	9	5	5	2	9	1	5	5	4	5	9	1	7	5	5	4	9	1	4	5	5	4	10	1	2	5	5	4	10	1	5	5	5	4	10
5	4	9	3	9	5	5	3	9	3	3	5	5	3	9	55	2	5	5	4	10	3	7	5	5	4	9	6	12	5	5	5	9	3	3	5	5	3	9
5	4	9	5	3	5	5	3	9	5	1	5	5	3	9	56	11	5	5	4	10	5	4	5	5	4	9	6	12	5	5	5	9	5	2	5	5	3	9
4	3	8	49	18	5	5	1	8	29	39	5	4	1	8	29	28	5	4	1	8	75	22	5	4	3	8	49	2	5	4	2	8	29	24	5	4	1	9
4	3	8	49	3	5	5	2	8	49	1	5	5	2	8	29	4	5	5	1	8	75	3	5	4	2	8	49	21	5	4	3	8	29	1	5	5	3	9
3	2	8	60	23	5	4	3	8	11	22	5	4	2	8	39	71	5	5	4	9	40	72	5	5	4	9	40	17	5	4	3	8	40	32	5	4	4	8
4	3	8	41	31	4	4	4	8	41	29	4	5	4	9	41	47	4	4	4	8	100	51	5	4	3	8	41	34	4	5	5	9	41	42	4	5	4	9
5	3	9	15	8	5	5	3	9	75	5	4	5	5	9	15	19	5	5	2	9	15	4	5	5	3	10	15	4	5	5	4	10	15	27	5	5	3	9
5	4	9	17	5	5	5	3	9	80	21	4	5	5	9	17	4	5	5	3	9	17	2	5	5	3	10	17	3	5	5	4	10	17	3	5	5	3	9
5	3	9	22	3	5	5	5	10	22	2	5	5	4	9	22	3	5	5	3	10	22	19	5	5	3	9	22	2	5	5	4	9	22	9	5	5	3	9
5	3	9	46	1	5	5	5	10	46	1	5	5	4	9	46	3	5	5	3	10	46	3	5	5	3	9	46	7	5	5	3	9	46	1	5	5	4	9
5	3	9	48	2	5	4	4	9	48	2	5	5	2	10	47	1	5	5	4	9	48	4	5	5	3	10	48	3	5	5	3	10	48	3	5	5	4	9
4	3	8	85	78	4	5	5	9	89	52	5	4	3	8	86	47	5	5	3	8	98	27	4	4	4	8	86	47	5	5	4	9	106	151	5	4	3	8
4	4	8	85	2	4	5	5	9	89	2	5	4	4	8	86	2	5	4	3	8	98	3	4	4	3	8	86	2	5	5	3	9	106	2	5	4	4	8
5	2	10	29	4	5	5	2	10	29	2	5	5	2	10	29	1	5	5	3	10	29	5	5	5	3	9	29	2	5	5	3	10	29	4	5	5	4	10
5	4	10	31	111	3	5	5	9	31	3	4	5	5	10	31	1	3	5	5	10	31	2	4	5	5	10	31	3	4	5	5	10	31	1	4	5	5	10
5	4	10	33	11	3	5	5	9	33	31	3	5	5	10	33	3	3	5	5	10	33	3	4	5	5	10	33	2	4	5	5	10	33	4	4	5	5	10

Bibliography

[noa,] Gartner Identifies the Top 10 Strategic Technology Trends for 2020.

[Abidin et al., 2017] Abidin, R. Z., Arshad, H., and Shukri, S. (2017). A Framework of Adaptive Multimodal Input for Location-Based Augmented Reality Application. *undefined*.

[Aigner et al., 2012] Aigner, R., Wigdor, D., Benko, H., Haller, M., Lindbauer, D., Ion, A., Zhao, S., and Koh, J. T. K. V. (2012). Understanding Mid-Air Hand Gestures: A Study of Human Preferences in Usage of Gesture Types for HCI.

[Aitken and Ross, 2019] Aitken, J. L. and Ross, H. A. (2019). *Mixed Reality Training Method: Performance Benefits for Routine Vehicle Maintenance Tasks*. Defence Technology Agency. Google-Books-ID: hC3SxgEACAAJ.

[Al Janabi et al., 2020] Al Janabi, H. F., Aydin, A., Palaneer, S., Macchione, N., Al-Jabir, A., Khan, M. S., Dasgupta, P., and Ahmed, K. (2020). Effectiveness of the HoloLens mixed-reality headset in minimally invasive surgery: a simulation-based feasibility study. *Surgical Endoscopy*, 34(3):1143–1149.

[Ali et al., 2019] Ali, A. X., Morris, M. R., and Wobbrock, J. O. (2019). Crowdlicit: A System for Conducting Distributed End-User Elicitation and Identification Studies. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–12, New York, NY, USA. Association for Computing Machinery.

[Azuma, 1997] Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.

[Billinghurst et al., 2015] Billinghurst, M., Clark, A., and Lee, G. (2015). A Survey of Augmented Reality. *Foundations and Trends® in Human-Computer Interaction*, 8:73–272.

[Cafaro et al., 2018] Cafaro, F., Lyons, L., and Antle, A. (2018). Framed Guessability: Improving the Discoverability of Gestures and Body Movements for Full-Body Interaction. pages 1–12.

[Chalhoub and Ayer, 2018] Chalhoub, J. and Ayer, S. K. (2018). Using Mixed Reality for electrical construction design communication. *Automation in Construction*, 86:1–10.

[Cometti et al., 2018] Cometti, C., Païzis, C., Casteleira, A., Pons, G., and Babault, N. (2018). Effects of mixed reality head-mounted glasses during 90 minutes of mental and manual tasks on cognitive and physiological functions. *PeerJ*, 6:e5847. Publisher: PeerJ Inc.

[Davis et al., 2014] Davis, S., Nesbitt, K., and Nalivaiko, E. (2014). A Systematic Review of Cybersickness. In *Proceedings of the 2014 Conference on Interactive Entertainment*, IE2014, pages 1–9, New York, NY, USA. Association for Computing Machinery.

[Duarte and Carriço, 2006] Duarte, C. and Carriço, L. (2006). A conceptual framework for developing adaptive multimodal applications. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, pages 132–139, New York, NY, USA. Association for Computing Machinery.

[Dumas et al., 2009] Dumas, B., Lalanne, D., and Oviatt, S. (2009). Multimodal Interfaces: A Survey of Principles, Models and Frameworks. volume 5440, pages 3–26.

[Espíndola et al., 2013] Espíndola, D. B., Fumagalli, L., Garetti, M., Pereira, C. E., Botelho, S. S. C., and Ventura Henriques, R. (2013). A model-based approach for data integration to improve maintenance management by mixed reality. *Computers in Industry*, 64(4):376–391.

[Feick et al., 2018] Feick, M., Tang, A., and Bateman, S. (2018). Mixed-Reality for Object-Focused Remote Collaboration. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*, pages 63–65, Berlin Germany. ACM.

[Fernández del Amo et al., 2018] Fernández del Amo, I., Erkoyuncu, J. A., Roy, R., and Wilding, S. (2018). Augmented Reality in Maintenance: An information-centred design framework. *Procedia Manufacturing*, 19:148–155.

[Fonnet et al., 2017] Fonnet, A., Alves, N., Sousa, N., Guevara, M., and Magalhães, L. (2017). Heritage BIM integration with mixed reality for building preventive maintenance. In *2017 24º Encontro Português de Computação Gráfica e Interação (EPCGI)*, pages 1–7.

[Gibby et al., 2019] Gibby, J. T., Swenson, S. A., Cvetko, S., Rao, R., and Javan, R. (2019). Head-mounted display augmented reality to guide pedicle screw placement utilizing computed tomography. *International Journal of Computer Assisted Radiology and Surgery*, 14(3):525–535.

[Gimeno et al., 2013] Gimeno, J., Morillo, P., Orduña, J. M., and Fernández, M. (2013). A new AR authoring tool using depth maps for industrial procedures. *Computers in Industry*, 64(9):1263–1271.

[Guna et al., 2014] Guna, J., Jakus, G., Pogacnik, M., Tomazic, S., and Sodnik, J. (2014). An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors (Basel, Switzerland)*, 14:3702–20.

[HTC,] HTC. The professional-grade VR headset | VIVE Pro France.

[Incekara et al., 2018] Incekara, F., Smits, M., Dirven, C., and Vincent, A. (2018). Clinical Feasibility of a Wearable Mixed-Reality Device in Neurosurgery. *World Neurosurgery*, 118:e422–e427.

[Jacucci et al., 2018] Jacucci, G., Andolina, S., Kalkhofen, D., Schmalstieg, D., Nurminen, A., Spagnolli, A., Gamberini, L., and Ruotsalo, T. (2018). Combining Intelligent Recommendation and Mixed Reality in Itineraries for Urban Exploration. *International SERIES on Information Systems and Management in Creative eMedia (CreMedia)*, (2017/2):18–23. Number: 2017/2.

[Juraschek et al., 2018] Juraschek, M., Büth, L., Posselt, G., and Herrmann, C. (2018). Mixed Reality in Learning Factories. *Procedia Manufacturing*, 23:153–158.

[Kaplan et al., 2021] Kaplan, A. D., Cruit, J., Endsley, M., Beers, S. M., Sawyer, B. D., and Hancock, P. A. (2021). The Effects of Virtual Reality, Augmented Reality, and Mixed Reality as Training Enhancement Methods: A Meta-Analysis. *Human Factors*, 63(4):706–726. Publisher: SAGE Publications Inc.

[Khronos, 2016] Khronos (2016). OpenXR - High-performance access to AR and VR —collectively known as XR— platforms and devices. Section: API.

[Kim et al., 2019] Kim, J., Agarwal, S., Marotta, K., Li, S., Leo, J., and Chau, D. H. (2019). Mixed Reality for Learning Programming. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, pages 574–579, New York, NY, USA. Association for Computing Machinery.

[Kong et al., 2011] Kong, J., Zhang, W., Yu, N., and Xia, X. (2011). Design of human-centric adaptive multimodal interfaces. *Int. J. Hum.-Comput. Stud.*, 69:854–869.

[Krings et al., 2020] Krings, S., Yigitbas, E., Jovanovikj, I., Sauer, S., and Engels, G. (2020). Development framework for context-aware augmented reality applications. In *Companion Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '20 Companion, pages 1–6, New York, NY, USA. Association for Computing Machinery.

[Leap,] Leap, M. Magic Leap 1 AR information and pricing | Magic Leap.

[LeapMotion, 2015] LeapMotion (2015). How Does the Leap Motion Controller Work?

[Lee and Billinghurst, 2013] Lee, G. A. and Billinghurst, M. (2013). A component based framework for mobile outdoor AR applications. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, SA '13, page 1, New York, NY, USA. Association for Computing Machinery.

[Lin et al., 2019] Lin, J., Harris-Adamson, C., and Rempel, D. (2019). The Design of Hand Gestures for Selecting Virtual Objects. *International Journal of Human–Computer Interaction*, 35(18):1729–1735. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2019.1571783.

[Link et al., 2016] Link, S., Barkschat, B., Zimmerer, C., Fischbach, M., Wiebusch, D., Lugrin, J.-L., and Latoschik, M. E. (2016). An intelligent multimodal mixed reality real-time strategy game. In *2016 IEEE Virtual Reality (VR)*, pages 223–224. ISSN: 2375-5334.

[Lynx,] Lynx. Lynx-R1 : Mixed Reality Standalone Headset.

[Madapana et al., 2018] Madapana, N., Gonzalez, G., Rodgers, R., Zhang, L., and Wachs, J. P. (2018). Gestures for Picture Archiving and Communication Systems (PACS) operation in the operating room: Is there any standard? *PLoS ONE*, 13(6):e0198092.

[Michael Kipp,] Michael Kipp. Leap Motion.

[Microsoft,] Microsoft. HoloLens 2—Overview, Features, and Specs | Microsoft HoloLens.

[Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems*, vol. E77-D, no. 12:1321–1329.

[Minkyung Lee,] Minkyung Lee. A usability study of multimodal input in an augmented reality environment | SpringerLink.

[Moniri et al., 2016] Moniri, M., Valcarcel, F. A. E., Merkel, D., Schuffert, W., and Schwartz, T. (2016). Hybrid team interaction in the mixed reality continuum. *VRST*.

[Morris et al., 2014] Morris, M. R., Danielescu, A., Drucker, S., Fisher, D., Lee, B., schraefel, m. c., and Wobbrock, J. O. (2014). Reducing legacy bias in gesture elicitation studies. *Interactions*, 21(3):40–45.

[Möller et al., 2014] Möller, A., Diewald, S., Roalter, L., and Kranz, M. (2014). Supporting Mobile Multimodal Interaction with a Rule-Based Framework.

[Novikov, 2019] Novikov, K. (2019). MIXED REALITY IN MAINTENANCE SVOČ – FST 2019.

[Park et al., 2020] Park, B. J., Hunt, S. J., Martin, C., Nadolski, G. J., Wood, B. J., and Gade, T. P. (2020). Augmented and Mixed Reality: Technologies for Enhancing the Future of IR. *Journal of Vascular and Interventional Radiology*, 31(7):1074–1082.

[Parveau and Adda, 2018] Parveau, M. and Adda, M. (2018). 3iVClass: a new classification method for Virtual, Augmented and Mixed Realities. *Procedia Computer Science*, 141:263–270.

[Parveau and Adda, 2020] Parveau, M. and Adda, M. (2020). Toward a user-centric classification scheme for extended reality paradigms. *Journal of Ambient Intelligence and Humanized Computing*, 11.

[Piumsomboon et al., 2013] Piumsomboon, T., Clark, A., Billinghurst, M., and Cockburn, A. (2013). User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts* *on Human Factors in Computing Systems*, CHI EA '13, pages 955–960, New York, NY, USA. Association for Computing Machinery.

[Qian et al., 2017] Qian, L., Azimi, E., Kazanzides, P., and Navab, N. (2017). Comprehensive Tracker Based Display Calibration for Holographic Optical See-Through Head-Mounted Display. *undefined*.

[Rae et al., 2018] Rae, E., Lasso, A., Holden, M. S., Morin, E., Levy, R., and Fichtinger, G. (2018). Neurosurgical burr hole placement using the Microsoft HoloLens. In *Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10576, pages 190–197. SPIE.

[Rekimoto and Nagao, 1995] Rekimoto, J. and Nagao, K. (1995). The World through the Computer: Computer Augmented Interaction with Real World Environments. *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*.

[Rokhsaritalemi et al., 2020] Rokhsaritalemi, S., Sadeghi-Niaraki, A., and Choi, S.-M. (2020). A Review on Mixed Reality: Current Trends, Challenges and Prospects. *Applied Sciences*, 10(2):636. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[Shao and Stanford, 2016] Shao, L. and Stanford, E. E. (2016). Hand movement and gesture recognition using Leap Motion Controller.

[Solórzano,] Solórzano, M. F. Mobile Multimodal Interaction: An Investigation and Implementation of Context-dependent Adaptation | WISE Lab.

[Speicher et al., 2019] Speicher, M., Hall, B. D., and Nebeling, M. (2019). What is Mixed Reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15. Association for Computing Machinery, New York, NY, USA.

[Stretton et al., 2018] Stretton, T., Cochrane, T., and Narayan, V. (2018). Exploring mobile mixed reality in healthcare higher education: A systematic review. *Research in Learning Technology*, 26.

[Suarez Fernandez et al., 2016] Suarez Fernandez, R., Sanchez-Lopez, J., Sampedro Pérez, C., Bavle, H., Molina, M., and Campoy, P. (2016). Natural user interfaces for human-drone multi-modal interaction. pages 1013–1022.

[Tsandilas, 2018] Tsandilas, T. (2018). Fallacies of Agreement: A Critical Review of Consensus Assessment Methods for Gesture Elicitation. *ACM Transactions on Computer-Human Interaction*, 25(3):18:1–18:49.

[UltraLeap, a] UltraLeap. API Overview — Leap Motion C# SDK v2.3 documentation.

[UltraLeap, b] UltraLeap. Ultraleap.

[Utzig et al., 2019] Utzig, S., Kaps, R., Azeem, S. M., and Gerndt, A. (2019). Augmented Reality for Remote Collaboration in Aircraft Maintenance Tasks. In *2019 IEEE Aerospace Conference*, pages 1–10. ISSN: 1095-323X.

[Varjo,] Varjo. Varjo XR-3 - The industry's highest resolution XR headset | Varjo.

[Vatavu, 2019] Vatavu, R.-D. (2019). The Dissimilarity-Consensus Approach to Agreement Analysis in Gesture Elicitation Studies. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–13, New York, NY, USA. Association for Computing Machinery.

[Vatavu and Wobbrock, 2015] Vatavu, R.-D. and Wobbrock, J. O. (2015). Formalizing Agreement Analysis for Elicitation Studies: New Measures, Significance Test, and Toolkit. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1325–1334, New York, NY, USA. Association for Computing Machinery.

[Vatavu and Wobbrock, 2016] Vatavu, R.-D. and Wobbrock, J. O. (2016). Between-Subjects Elicitation Studies: Formalization and Tool Support. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3390–3402, New York, NY, USA. Association for Computing Machinery.

[Villarreal-Narvaez et al., 2020] Villarreal-Narvaez, S., Vanderdonckt, J., Vatavu, R.-D., and Wobbrock, J. O. (2020). A Systematic Review of Gesture Elicitation Studies: What Can We Learn from 216 Studies? In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 855–872. Association for Computing Machinery, New York, NY, USA.

[Vysocky et al., 2020] Vysocky, A., Grushko, S., Oščádal, P., Kot, T., Babjak, J., Jánoš, R., Sukop, M., and Bobovský, Z. (2020). Analysis of Precision and Stability of Hand Tracking with Leap Motion Sensor. *Sensors*, 20:4088.

[Weichert et al., 2013] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors (Basel, Switzerland)*, 13:6380–6393.

[Wikipedia, 2022] Wikipedia (2022). List of game engines. Page Version ID: 1083849985.

[Wobbrock et al., 2005] Wobbrock, J. O., Aung, H. H., Rothrock, B., and Myers, B. A. (2005). Maximizing the guessability of symbolic input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1869–1872, New York, NY, USA. Association for Computing Machinery.

[Wobbrock et al., 2009] Wobbrock, J. O., Morris, M. R., and Wilson, A. D. (2009). User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1083–1092, New York, NY, USA. Association for Computing Machinery.

[Wu et al., 2019] Wu, H., Yang, L., Fu, S., and Zhang, X. (2019). Beyond remote control: Exploring natural gesture inputs for smart TV systems. *J. Ambient Intell. Smart Environ.*

[Zhu et al., 2013] Zhu, J., Ong, S. K., and Nee, A. Y. C. (2013). An authorable context-aware augmented reality system to assist the maintenance technicians. *Scopus*. Accepted: 2014-06-17T06:11:18Z.

[Zimmer et al., 2017] Zimmer, C., Bertram, M., Büntig, F., Drochtert, D., and Geiger, C. (2017). Mobile augmented reality illustrations that entertain and inform: design and implementation issues with the hololens. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pages 1–7, New York, NY, USA. Association for Computing Machinery.